University of Alberta

Detection of Abrupt Changes and Industrial Applications

by

Ashish Malhotra

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

in

Process Control

Department of Department of Chemical and Materials Engineering

Edmonton, Alberta
Fall 2000

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-59842-X

Canada

University of Alberta

Library Release Form

Name of Author: Ashish Malhotra

Title of Thesis: Detection of Abrupt Changes and Industrial Applications

Degree: Master of Science

Year this Degree Granted: 2000

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Ashish Malhotra

Ashish Malhotra
CME 536
University of Alberta
Edmonton, AB
Canada, T6G 2G6

Date: 2/Oct/2000

If a man will begin with certainties, he will end in doubts; but if he will be content to begin with doubts, he will end in certainties.
Francis Bacon (1561-1626)

Abstract

Fault detection is fast emerging as an important tool in the field of process control. The primary objective of fault detection is to detect both fast and slow undesirable changes in a process at an early stage. This early detection has a potential of preventing loss of production and equipment damage due to these undesirable changes, thus reducing process downtime. This work details the implementation of some parametric fault detection techniques on data from chemical process industries. A parametric fault detection approach which is handled in depth in this thesis is the local approach. This approach developed by Basseville and Benvensite (1987) offers a computationally inexpensive way to attain the objective of monitoring changes in model parameters. Work done in the literature on this approach to increase its applicability is presented in a condensed format. The results from the application of this approach to industrial data are compared with results obtained from some other fault detection methods (singular value decomposition and PCA). The local approach is then coupled with other techniques to arrive at innovative applications. For example it is combined with modeling techniques such as least squares, output error method and prediction error method to derive detection algorithms with different applicability. The local approach is also coupled with control relevant identification to devise a control relevant model validation algorithm.

Control relevant identification is to search for appropriate data prefilters before applying identification schemes like the prediction error method, such that the identified model is most suitable for controller design. Once a model is obtained, it is equally important to continuously validate the model to maintain the model quality. The reason for deriving a control relevant model validation algorithm comes from the observation that at times models used in model predictive control do not pass the 'rigorous' model validation test but the control systems which use these models still gives good performance. This raises the question of what exactly is a 'good' process model and also whether the regular validation tests which check the prediction residuals are too restrictive. The local approach is one of the many powerful statistical tools available in the literature for detection of abrupt parameter

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Detection of Abrupt Changes and Industrial Applications** submitted by Ashish Malhotra in partial fulfillment of the requirements for the degree of **Master of Science** in *Process Control.*

B. Huang (Supervisor)

T. Chen

E.S. Meadows

Date: 2/OCT/2000

To my parents

Acknowledgements

I would like to express my gratitude to Dr. Biao Huang for his supervision of this work. This work would not have materialized without his constant guidance and support. His deep insight, fresh ideas and constant encouragement were instrumental in driving me on. I would also like to thank my Profs. Shah, Chen, Huang and Weins for teaching me the fundamentals of control, Identification and statistics which proved to be useful tools in doing my research.

I had the pleasure of spending the last two years in the computer process control group here at the University of Alberta. I am grateful to the challenging, enjoyable and learning interactions with fellow students, professors and post doctoral fellow's here. Every one has been most helpful and encouraging. A word of thanks also to the faculty and staff of the *Department of Chemical and Materials Engineering* for the help and resources. The financial support of *Natural Sciences and Engineering Research Council of Canada* and *Syncrude Canada Ltd.* is gratefully acknowledged. A special thanks to the engineers at Syncrude Upgrading, Ft. McMurray for helping me in my projects there and teaching me the practical aspects of controls.

Lastly to all my friends who have made the last two years most enjoyable, you are the best.

changes and can be used for model validation by monitoring the system for parametric changes. The challenge though is that the algorithm is sensitive to any small change of parameters, which may not be desired in practice since not all parameter changes can affect control performance. Data prefiltering can be utilized to arrive at a control relevant local detection algorithm with the objective of making the algorithm robust to changes which do not affect controller performance. In this work, model predictive control relevant data prefilters are proposed. These filters can significantly reduce the unnecessary alarms and increase sensitivity to changes that truly affect controller performance.

The local approach is also coupled with input independent state estimation algorithm to arrive at an input independent fault detection approach with implementation to sensor decalibration. The main stumbling block to the sensor decalibration problem is that there is no direct knowledge of the inputs to the sensor and the decision on whether the sensor is at fault or not is done using only the outputs from the sensor. Here some recent results in Kalman filtering without the knowledge of inputs (Darouch 1997, Kitanidis 1987) are utilized to derive a local approach based sensor monitoring algorithm. This algorithm can be used to detect sensor fault or decalibration without the knowledge of the inputs.

Contents

1 Introduction 1

1.1 Scope of this thesis and research objectives 4

1.2 About this thesis 5

2 The Local Approach and Its Industrial Applications 6

2.1 Introduction 6

2.2 The local approach 7

2.2.1 Understanding the local approach 8

2.2.2 Local approach based monitoring algorithm 10

2.3 Robustness issues 11

2.4 Dealing with non-linear dynamic systems 14

2.4.1 Representation of nonlinear systems by differential algebraic equations (DAEs) 14

2.4.2 Elimination of unknown variables 15

2.4.3 Residual evaluation for fault detection 16

2.5 Isolation 19

2.5.1 Sensitivity test 19

2.5.2 Minmax test 20

2.6 Simulation examples 21

2.7 Industrial applications 24

2.7.1 Process monitoring 24

2.7.2 Process monitoring based on singular value decomposition 26

2.7.3 Monitoring Based on local approach and least squares modeling . . . 29

2.7.4 Comparison of the local approach with PCA 34

2.8 Conclusions 36

3 Control Relevant Model Validation and Its Application 37

3.1 Introduction 37

3.2 Data prefiltering for control relevant identification 38

3.2.1 Data prefiltering for optimal k-step ahead prediction model identification 40

3.2.2 Data prefiltering for optimal multistep prediction model identification 42
3.3 KPEM and MPEM relevant model validation 44
3.4 Simulation 48
3.5 Industrial applications 52
3.6 Conclusion 55

4 Monitoring of Process Gain Decalibration Without the Knowledge of Inputs 60
4.1 Introduction 60
4.2 Sensor decalibration monitoring for slow-changing inputs 63
4.2.1 Monitoring algorithm using local approach 63
4.2.2 Results 64
4.3 Sensor decalibration monitoring for fast-changing input 66
4.3.1 Monitoring algorithm using input independent Kalman filtering . . . 66
4.3.2 Simulation and pilot plant case study 70
4.4 Conclusions 76

5 Summary and Conclusions 77

Bibliography 80

A Graphical User Interface for predictive control relevant model validation 83
A.1 Getting started 83
A.2 Inputs to the GUI 83
A.3 Outputs of the GUI 87

List of Figures

1.1 Classification of faults 3

2.1 spectra plot of the AR processes given in table 2.2, the solid line corresponds to Θ_0, the dashed line Θ_1, the dotted line Θ_2 and the dash dotted line is the least square identified AR(9) model. 23

2.2 spectra plot of the monitoring models and the actual process, the solid line corresponds to actual process, the dashed line the AR(2) model (autoregressive model of order of two), the dotted line the AR(3) model and the dash dotted line is the AR(4) model. 23

2.3 The simplified flow diagram of the industrial distillation column used for the mass balance analysis showing the flow meters. 28

2.4 (top-left) two variables representative of the flooding (top-right) SVD monitoring plot without the delay being taken into account (bottom) SVD monitoring plot with some approximate delay structure being taken into account - the dotted line shows the threshold. 29

2.5 Industrial Fluidized Coker with all controlled variables, manipulated variables, disturbance variables. 30

2.6 (top) two representative variables of a possible faulty node in the Fluidized Coker and (bottom) the resulting SVD monitoring plot - the dotted line shows the threshold. 31

2.7 (top-left) two variables representative of the flooding (top-right) χ^2 plot without the delay being taken into account (bottom) χ^2 plot with some approximate delay structure being taken into account - the dotted line shows the threshold. 33

2.8 (top-left) two variables representative of the flooding (top-right) χ^2 plot without the delay being taken into account (bottom) χ^2 plot with some approximate delay structure being taken into account - the dotted line shows the threshold. 34

2.9 SPE plot from PCA on flooding data. 35

2.10 Hotelling's T^2 plot from PCA on flooding data. 35

3.1 Basic structure of a Model Predictive Controller. 38

3.2 Time domain trajectory of the actions of a Model Predictive Controller. . . 39
3.3 normalized amplitude Bode plot of KPEM filter (using 5,10,15 and 50 step). 49
3.4 normalized amplitude Bode plot of MPEM filter (using 5,10,15 and 50 step). 49
3.5 plot of χ^2 test using the KOEM based detection algorithm. 50
3.6 plot of χ^2 test using the MOEM based detection algorithm. 51
3.7 plot of χ^2 test using the KPEM based detection algorithm. 51
3.8 χ^2 test using the MPEM based detection algorithm. 52
3.9 χ^2 test using the KOEM filter - Fluidized Coker case study. 53
3.10 χ^2 test using the MOEM filter - Fluidized Coker case study. 54
3.11 χ^2 test using the KPEM filter - Fluidized Coker case study. 55
3.12 χ^2 test using the MPEM filter - Fluidized Coker case study. 56
3.13 χ^2 plotted against the prediction horizons - Hydrogen plant case study - cv_1. 56
3.14 χ^2 plotted against the prediction horizons - Hydrogen plant case study - cv_6. 57
3.15 Result of isolation tests carried on cv_6. 58
3.16 χ^2 test scores plotted against the prediction horizons for cv_5. 58
3.17 Result of Isolation tests carried on cv_5 for ten prediction horizons. 59

4.1 The inherent difficulty in sensor monitoring as compared to process or control loop monitoring 61
4.2 Combined Gain Monitoring 62
4.3 calculated χ^2 plot of a twenty percent change in the sensor gain at around one thousand sampling instances - the threshold is shown as a dotted line. . 65
4.4 calculated χ^2 plot of a forty percent change in the sensor gain at around one thousand sampling instances - the threshold is shown as a dotted line. . . . 66
4.5 (left) time series plot of historized data and (right) the resulting χ^2 plot from the monitoring algorithm. 67
4.6 schematic of the pilot plant set up 72
4.7 plot of the outputs along with the estimates from the input independent Kalman filter. The solid line corresponds to actual outputs and the dotted line the estimated outputs. 73
4.8 plot of the actual outputs (solid line) along with the outputs with 20 % gain change (dotted line) at the two hundredth sampling instance. 74
4.9 plot of the outputs with a 20 % change in the sensor gain at the two hundredth sampling instance (solid line) along with the estimates from the input independent Kalman filter (dotted line). 74
4.10 plot of the outputs (solid line) along with the estimates from the input independent Kalman filter (dotted line) for routine data. 75
4.11 plot of the outputs (solid line) along with the outputs with 20 % change in the sensor gain at the two hundredth sampling instance (dotted line). . . . 75

4.12 plot of the outputs (solid line) with a 20 % change in the sensor gain at the two hundredth sampling instance along with the estimates from the input independent Kalman filter (dotted line). 76

A.1 Graphical User Interface, Inputs 84
A.2 Graphical User Interface, Inputing the models 85
A.3 load model file dialogue box 86
A.4 Graphical User Interface, Outputs 87

List of Tables

2.1 *Results of local approach detection algorithm on simulation runs* 21

2.2 *AR process parameters chosen for the simulation with the value of the parameters for two faulty nodes* 22

2.3 *AR model parameters chosen for monitoring* 24

2.4 *Calculated χ^2 values for different monitoring $\theta's$ with no change Θ_0* 24

2.5 *Calculated χ^2 values for different monitoring $\theta's$ with change from Θ_0 to Θ_1* 25

2.6 *Calculated χ^2 values for different monitoring $\theta's$ with change from Θ_0 to Θ_2* 25

4.1 *Results of sensor decalibration algorithm on simulated data.* 71

4.2 *Results of Sensor decalibration algorithm with different changes in the sensor gain at the two hundredth sampling instance* 75

Chapter 1

Introduction

The field of fault detection and isolation has attracted much interest from the control community over the last two decades. The attention from the academia and industry for such a multidisciplinary and diverse field has led to its rapid growth and it is gradually becoming a mature discipline. The basic premise of fault detection is similar to the classical model validation problem. To put the problem involved in very simple terms most fault detection approaches require a parametric or nonparametric representation of the system. This representation is of the system in its normal or fault free state. The future data from the system is then checked against this representation to see if the relationship still holds. This checking or monitoring can be done over batches of data (offline monitoring) or on a continuous basis (online monitoring). Parametric fault detection and model validation are in fact interchangeable problems as both deal with monitoring the change in the parameters which describe the system. From this general description of the detection problem various different approaches take stem and branch out, the difference lying in the representative model used for monitoring or in the tools used for the comparison of the future data with the model.

In most of the literature discussed in this work undesirable changes in a process are referred to as faults (Willsky 1976, Isermann 1984, Gertler 1988, Basseville 1988, Frank 1990). This covers malfunctions, break downs, failures and other events which disrupt the plant and take it away from its normal operating state. These faults or undesirable changes, are broadly classified as the fast changes and slow changes. Fast changes are changes which are fast in comparison to the sampling rate and typically have small magnitude - like failure of some instrument, tripping of a pump etc. The slow changes are changes linked to slow degradation of the system - like slow change in the heat transfer coefficient of a heat exchanger due to fouling or the gradual change in the mass transfer coefficients of a distillation column due to gradual plugging of the sieve trays. These changes often escape detection until it is too late and may result in a loss in productivity, in extreme cases may even force an unscheduled shutdown of the process. Timely detection however may help in facilitating preventive maintenance. Early fault detection can be used to change

maintenance from being periodic calendar based to being condition-based or preventive maintenance (Zhang *et al.* 1994). The implementation of this objective of early detection has been approached in various ways in literature by using different tools like neural networks (Marcu 1994), wavelets (Aravena 1996), singular value decomposition (Maryak *et al.* 1997), Generalized likelihood ratio test (Apley 1994) among others.

These approaches have tremendous potential for application in chemical process industries. Chemical industries normally operate under continuous conditions and typically handle large volume throughput. For example, Syncrude Canada facility at Ft. McMurray, Alberta produced an overall annual shipment of 75.7 million barrels of synthetic crude oil in 1997 (Syncrude Canada Ltd. 1997, Annual Report), Nova chemical produced 2.6 billion pounds of polyethylene in 1998. Faults in the chemical industries which handle such large volumes would result in deviation of the process from the normal operation trajectory causing degradation of product quality and production of large amounts of off-spec product. The large volumes and continuous nature of operation also means that the start up and shut down typically are very time consuming and costly procedures. So the extreme case of a fault causing a shut down equates to great losses due to process downtime and subsequent startup. In this regard the ability to predict an undesirable change in the process would help to increase profitability by helping in maintaining product quality and by reducing process down time.

Faults are also frequently classified as additive and non-additive faults. These distinction is best described in figure 1.1 (Basseville 1998). Here Y is the observed data, U is the known inputs, W_s the unknown inputs (the unknown disturbances and input noise). W_o is the output noise. The first type of faults is shown as γ_o and acts at the output of the system in an additive manner, i.e. it only affects the mean value of the output Y or in simple terms it adds to the output Y. This can be visualized as sensor fault (for *e.g.* a bias in sensor measurements). The second type of faults denoted by γ_i acts at the input of the system again in an additive manner. For a linear system this type of fault would also affect the mean value of the output Y. This can be visualized as the actuator faults. The third type of faults are the faults denoted as the 'change in θ'. These affect the system generating mechanism and the affect of these changes on the output is not just a mean change, but a change in the variance, correlations and other higher order characteristics. Hence this type of faults are called non-additive faults or for linear systems multiplicative faults (Basseville and Nikiforov 1993). These types of faults can be visualized as component or system faults (Patton *et al.* 1989). Isolation of the faulty parameter is the next logical step in the fault detection methodology. Here the basic idea is that once the detection algorithm suggests that a fault has occured, i.e the parameterization used does not describe the system properly, it is important to pinpoint the fault to the erring parameter. Some work on this has been done in the literature, for example the sensitivity and the minmax test in Zhang *et al.* (1998) but this field is still largely unexplored.

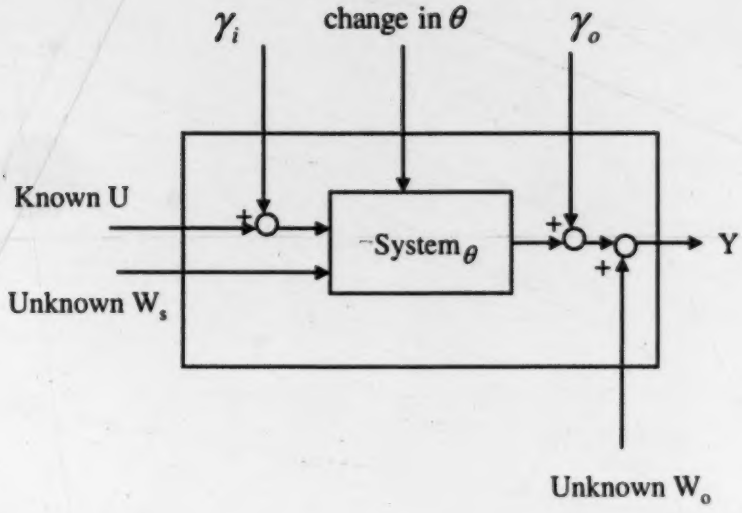

Figure 1.1: Classification of faults

As mentioned earlier the root of the detection problem and the validation problem is similar - monitoring the change in parameters of some representation of the process under consideration. Hence the vast array of statistical tools developed for fault detection can be used for model validation too, and vice versa. With the wide spread use of model predictive controllers (MPC) in the industry there is a need to develop tools to improve and assess the performance of these advanced controllers. This is by no means a simple task due to the multivariate nature of the controller and the complexity of the optimizer used for designing the control moves. A good way to assess and possibly improve the performance of the MPC is to develop tools for validation of the model used by the controller. This model forms a very important part of the MPC as the predictions from this model are used by the optimizer to decide the future control moves (figure 3.1 and 3.2). It is seen in practice that the conventional techniques for model validation such as the residual whiteness test can be too restrictive or too loose a condition for a model which is to be used for model predictive control. Models are seen to fail these conventional validation tests but still give good controller performance. Hence there is a need for an algorithm which validates the model with respect to the role the model has to perform. For example in model predictive controller a model has to generate predictions over a horizon, so in the validation strategy the model should only be checked to see if it can give good or optimal prediction over that horizon over which the controller operates. The validation tool should 'look' through the

eyes of a predictive controller while deciding if the model is good or not.

Sensors form an important link in control strategies, and yet the literature on the monitoring of sensors for early fault detection is pretty sparse. If we compare this to the mature field of process or controller performance monitoring we notice that the main stumbling block is that in the case of the sensor monitoring the inputs to the sensor are unknown. While in process or controller monitoring the relation between the inputs and outputs is checked to see if it can still be represented by the nominal model, in the case of sensor fault or decalibration monitoring only the output from the sensor is known. Hence the monitoring algorithm relies only on the output from the sensor, i.e the process measurements for making the decision. Hence there is a need for devising a detection strategy which would take into account such cases where the inputs are not known. Then the problem of sensor monitoring can be solved using such a detection strategy.

1.1 Scope of this thesis and research objectives

In this work the following research objectives have been considered:

1. Fault detection:
   The initial thrust of this work is to apply the statistical change detection approaches which are fast gathering interest in control literature to some industrial chemical processes. Here specific attention is given to the local approach of change detection, an approach which is used to detect non-additive or process abrupt changes which are small in magnitude.

2. Control relevant model validation:
   As explained before there is a need for developing tools for validating models for model predictive controllers. The validation strategy should detects only those parametric changes in the model which would affect the performance of a model based predictive controller. The local approach can be used for such detection of parameter changes, but the sensitivity of this approach can be detrimental in this case as all small changes may not actually affect the control performance. Data prefiltering can offer interesting solutions as weightings can be placed on those frequencies which would affect controller performance. So the validation strategy thus evolved using weightings on the frequencies of interest would only be sensitive to those changes that would affect the controller performance.

3. Input independent fault detection using Kalman filtering:
   Sensor decalibration and failure is a very common and serious problem in chemical process industries, where the number of sensors may run in thousands. As explained before the issue of lack of knowledge of inputs has to be tackled before a robust algorithm for sensor decalibration monitoring can be arrived upon. There exists some

work in the literature on minimum variance unbiased Kalman filters which do not require the knowledge of the inputs (Darouch 1997). Such work can be exploited to devise algorithms for applications such as the sensor monitoring without the knowledge of inputs. This estimators can then be used with the local approach to arrive at a sensor monitoring algorithm.

1.2 About this thesis

The organization of this thesis is as follows:

Chapter 2 deals with the local approach for fault detection and isolation. Here the approach is introduced using simple examples. Work done using the local approach in the literature is briefly explained. The approach is used along with different model formulation to increase its applicability. Several simulations and applications on data from industrial chemical processes are presented. Comparison results with singular value decomposition for fault detection (similar to last principal component analysis) and the PCA analysis are presented.

Chapter 3 deals with control relevant model validation. It gives a brief overview of model predictive controllers. Here the formulation of control relevant identification functions using data prefiltering is explained in detail. The data prefiltering derived in the identification part is used with the local approach to obtain a control relevant model validation algorithm. Simulation examples and application of the algorithm on industrial data from chemical processes have been presented.

Chapter 4 deals with monitoring of gain decalibration problems without the knowledge of inputs. This chapter details the application of input independent Kalman filtering to fault detection, specifically to the problem of sensor decalibration. The sensor decalibration and monitoring problem is introduced in this section. A solution from the literature which uses the local approach with ARIMA modeling is analyzed. As an alternative a detection algorithm is proposed which uses input independent Kalman filtering along with the local approach to handle cases of change detection without the knowledge of inputs. Simulations and experimental results conducted on a pilot plant are presented.

Conclusions are discussed in Chapter 5. Here a summary of the contributions of this thesis is presented.

Chapter 2

The Local Approach and Its Industrial Applications

2.1 Introduction

Fault detection and isolation aims at early detection of undesirable process changes or malfunctions (Willsky 1976, Isermann 1984, Gertler 1988, Basseville 1988, Frank 1990). If this objective of early detection can be achieved inexpensively and timely, then it can be easily applied to real time process monitoring. The successful implementation of such a continuous online approach can provide means to detect small deviations at an early stage. This can facilitate timely action and prevent damage to the system or downtime.

In the literature cited above faults (or malfunctions, undesirable process changes) are broadly classified as measurement faults, additive faults and non-additive or multiplicative faults. While measurement faults and additive faults can show up as a difference between the actual process output and the measured output the multiplicative or non-additive faults are difficult to detect by normal means of inspection (refer chapter 1). These faults can be visualized as changes acting through the process and hence affecting the 'generating mechanism' itself (like changes in the state transition matrix) (Basseville 1998). These non-additive faults can generally be represented as changes of plant parameters from their nominal or safe value (Gertler 1988, Basseville and Nikiforov 1993, Basseville 1998, Patton *et al.* 1989).

The actual computation and the strategy of most fault detection algorithms is generally split into two parts, namely, generation of residuals and residual evaluation. Ideally the residuals generated for the purpose of fault diagnosis should be close to zero for normal plant operation and non-zero for faulty operation. The residuals should have favorable properties like minimal sensitivity to disturbances and maximum sensitivity to faults. Residual evaluation is making decisions based on these residuals. A Statistical approach called the Local Approach (Benveniste *et al.* 1987, Basseville 1998), provides ways in which this objective (of residual generation and evaluation) can be attained in a neat, computationally inexpensive manner. This approach aims at early detection and isolation of small undesirable

non-additive changes in a process.

This chapter aims at giving a comprehensive overview of the local approach. Section 2.2 gives a brief background on the approach along with a summary of the algorithm. Section 2.3 discusses the work of Zhang *et al.* (1994) regarding the robustness issues related to the local approach. It is shown in this work by Zhang and his co-workers, how the initial limitations on the approach can be relaxed, it is an exercise which broadens the area of applicability of the approach. Section 2.4 deals with the work of Zhang *et al.* (1998) regarding the application of the local approach to nonlinear dynamic systems using some new results in the area of parameter identifiability (Ljung and Glad 1994). Section 2.5 gives the algorithm for isolation of faults. Isolation of the particular part of the parameter vector which causes undesirable changes is an important step which can help in pin pointing the fault. Section 2.6 gives a simulation example. section 2.7 deals with industrial applications of the local approach. Here the results are also compared with results obtained for the same industrial data using PCA for monitoring. Lastly section 2.8 gives the conclusion and summary.

2.2 The local approach

Basseville and co-workers started work on monitoring plant parametric changes (Basseville 1983, Benveniste *et al.* 1987) from which they developed a procedure for the detection and isolation of abrupt change in the parameters of a process (multiplicative or non-additive faults). This approach provides tools by which the detection problem of monitoring the change in the parameters of a stochastic process can be simplified to the problem of monitoring the mean of a Gaussian vector (Benveniste *et al.* 1987, LeCam 1986, Hall and Mathiason 1990). The tests used for change detection is the generalized likelihood ratio test. This test is nonlinear and is linearized over a local point (hence the name local approach).

A fair amount of work has been done in the topic of fault diagnosis using the local approach ever since. Robustness issues for the local approach are explored by Zhang *et al.* (1994). In this work issues such as effect of model plant mismatch and underparametrization on the detection algorithm are discussed. It is shown in this work that this approach has a wide applicability as it can deal with model reduction and bias in identification. This approach has also applied to monitoring of non-linear models (Zhang *et al.* 1998). Monitoring of non-linear models based on engineering principals which are usually represented by differential algebraic equations is targeted in this work. The differential equations are converted to an input output representation of the system using recent results in the area of identification (Ljung and Glad 1994).

The local approach has been applied to far reaching and challenging fields like monitoring of gas turbines, oil rigs vibration, emission from automobile catalytic converters (Basseville 1998), bioreactors (Zhang *et al.* 1998) etc. A variety of objectives can be attained in the

framework of this approach - such as model validation (Zhang *et al.* 1994), detection of sensor decalibration (O'Reilly 1998), process monitoring among other applications.

The formal problem statement for the local approach, can be written as an hypothesis test (Benveniste *et al.* 1987) :

$$\mathcal{H}_0 : \theta = \theta_0 \tag{2.1}$$

against

$$\mathcal{H}_1 : \theta = \theta_0 + \frac{\eta}{\sqrt{N}} \tag{2.2}$$

here the system is parameterized by the vector θ, with θ_0 representing the nominal value identified over a nominal or safe run of the system. This hypothesis test means that if there is no fault the present parameter vector will be the same as the nominal parameter vector, and if there is a fault the present parameter vector will be different from the nominal parameter vector. η is an arbitrary unknown vector having the same dimension as θ which signifies that the change of the system parameter from the nominal parameter can be in any arbitrary direction i.e. along any element of the parameter vector θ.

The local approach deals with the problem of change detection hypothesis test (equations 2.1 and 2.2) by transforming them to the problem of monitoring the mean of a Gaussian vector through the local approach.

2.2.1 Understanding the local approach

To understand the local approach consider the case of a stochastic process represented by a scalar linear regression of the form (Benveniste *et al.* 1987, Zhang *et al.* 1994):

$$y_k = \phi_k^T \theta^* + \epsilon_k \tag{2.3}$$

where y_k is the output signal, ϵ_k is a white noise sequence with Gaussian distribution and variance σ^2, θ^* is the actual process parameter and ϕ_k the regressor vector consisting of the exogenous measurable or known inputs. θ^* can be replaced by its nominal approximation θ_0. In this case the prediction errors or residual will be given by:

$$\hat{e}_k = y_k - \phi_k^T \theta_0 = \phi_k^T (\theta^* - \theta_0) + \epsilon_k \tag{2.4}$$

If there is any abrupt change in the parameters, the change can be defined as

$$\Delta\theta = (\theta - \theta^*)\sqrt{N} \tag{2.5}$$

where N is the length of the data used, the prediction error in this case becomes:

$$\begin{aligned} \hat{e}_k &= y_k - \phi_k^T \theta_0 \\ &= \phi_k^T \theta - \phi_k^T \theta_0 + \epsilon_k \end{aligned}$$

(here the parameter vector θ represents the system over which the errors are being calculated. This can be different from θ^* or θ_0.)
By adding and subtracting by $-\phi_k^T\theta^*$ and rearranging the terms we get:

$$\hat{e}_k = \phi_k^T \frac{\Delta\theta}{\sqrt{N}} + \phi_k^T(\theta^* - \theta_0) + \epsilon_k$$

The change can be in any of the parameters, i.e. any of the elements of $\Delta\theta$ can be non zero. Now we define

$$\begin{aligned} H(\phi_k, \theta_0) &= \phi_k \hat{e}_k \\ &= \phi_k\phi_k^T \frac{\Delta\theta}{\sqrt{N}} + \phi_k\phi_k^T(\theta^* - \theta_0) + \phi_k\epsilon_k \end{aligned} \tag{2.6}$$

and the cumulative sum of this quantity will be:

$$\begin{aligned} \xi_N(\theta_0) &= \frac{1}{\sqrt{N}}\Sigma_{k=1}^N H(\phi_k, \theta_0) \\ &= \frac{\Sigma_{k=1}^N \phi_k\phi_k^T}{N}\Delta\theta + \frac{\Sigma_{k=1}^N \phi_k\phi_k^T}{\sqrt{N}}(\theta^* - \theta_0) + \frac{\Sigma_{k=1}^N \phi_k\epsilon_k}{\sqrt{N}} \\ &= (a) + (b) + (c) \end{aligned}$$

the term (b) will vanish if we assume that a infinite (or large enough) data set has been used for the identification of the nominal parameter ($\theta_0 \rightarrow \theta^*$). Since both ϕ_k and ϵ_k are uncorrelated, $\phi_k\epsilon_k$ will also be zero mean and (c) will converge to some zero mean Gaussian distribution which will not depend on the true system (since both ϕ_k and ϵ_k are exogenous). If we consider $\Delta\theta$ to be fixed:

$$\text{Cov}\phi = \text{Covariance}(\phi_k) = \frac{\Sigma_{k=1}^N \phi_k\phi_k^T}{N} \in \Re^{n\times n} \tag{2.7}$$

by Central Limit theorem:

$$\xi_N(\theta_0) \sim \mathcal{N}(\text{Cov}\phi\Delta\theta, R) \tag{2.8}$$

where R is the asymptotic covariance matrix of (c). From equation 2.8 it is clear that a change in the system parameters will only be reflected in the mean of $\xi_N(\theta_0)$ and not in its covariance matrix, since $\Delta\theta$ term only figures in the mean of $\xi_N(\theta_0)$. Hence change in the parameter vector can detected as change in the mean of $\xi_N(\theta_0)$. In literature on the local approach $H(\theta_0, \phi_k)$ is referred to as the primary residual and $\xi_N(\theta_0)$ as the normalized residual.

Although the mathematical foundation is laid here for detection of abrupt change using the local approach, the challenge in practice is to find appropriate normalized residuals and primary residuals that are capable of detecting small changes while being *robust* to changes that are irrelevant to certain performance objectives.

2.2.2 Local approach based monitoring algorithm

Using the general example discussed in the previous section the monitoring algorithm can be summarized as follows:
A statistic $\xi_N(\theta_0)$ given by

$$\xi_N(\theta_0) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} H(\theta_0, z_t^{(\theta)})$$

is defined as a normalized residual if

$$E[H(\theta_0, z_t^{(\theta)})] = 0 \qquad \text{for } \theta = \theta_0 \tag{2.9}$$

and

$$E[H(\theta_0, z_t^{(\theta)})] \neq 0 \qquad \text{for } \theta \in \omega(\theta_0) \backslash \theta_0 \tag{2.10}$$

where $\omega(\theta_0)\backslash\theta_0$ reads as a neighborhood of θ_0 exclusive of θ_0. $H(\theta_0, z_t^{(\theta)})$ is defined as the primary residual, where $z_t^{(\theta)} = [u(t), y(t)]^T$. The θ notation has been put on z vector to make it clear that the information of the current system parameter vector θ (for comparison with the nominal vector θ_0) is obtained from this data vector z. Note that the residuals defined here are different from the regular prediction residuals, and are asymptotically sufficient statistics (Basseville 1998). The empirical version of equations 2.9 and 2.10 may be written as

$$\frac{1}{N} \sum_{t=1}^{N} H(\theta_0, z_t^{(\theta)}) = 0 \qquad \text{for } \theta = \theta_0 \tag{2.11}$$

and

$$\frac{1}{N} \sum_{t=1}^{N} H(\theta_0, z_t^{(\theta)}) \neq 0 \qquad \text{for } \theta \in \omega(\theta_0) \backslash \theta_0 \tag{2.12}$$

We will now refer to $z_t^{(\theta)}$ as z_t for compactness. If we can find such residuals, then for sufficient large N, the following results hold

$$\xi_N(\theta_0) \sim \mathcal{N}(0, \Sigma(\theta_0)) \qquad \text{under } H_0$$
$$\xi_N(\theta_0) \sim \mathcal{N}(-M(\theta_0)\eta, \Sigma(\theta_0)) \qquad \text{under } H_1$$

where

$$M(\theta_0) = (\frac{\partial}{\partial \theta} H(\theta_0, z_t)|_{\theta=\theta_0}) \tag{2.13}$$

$$\Sigma(\theta_0) = \sum_{t=-\infty}^{\infty} \text{Cov}(H(\theta_0, z_t), H(\theta_0, z_t)) \tag{2.14}$$

In practice, the empirical version of equation(2.13) may be written as

$$M(\theta_0) \approx \frac{\partial}{\partial\theta}[\frac{1}{N}\sum_{t=1}^{N} H(\theta_0, z_t)]_{\theta=\theta_0} \tag{2.15}$$

and $\Sigma(\theta_0)$ may be approximated by (Zhang *et al.* 1998)

$$\begin{aligned}\Sigma(\theta_0) \approx & \frac{1}{N}\sum_{t=1}^{N} H(\theta_0, z_t)H^T(\theta_0, z_t) + \\ & + \sum_{i=1}^{I} \frac{1}{N-i}\sum_{t=1}^{N-i}(H(\theta_0, z_t)H^T(\theta_0, z_{t+i}) + \\ & + H(\theta_0, z_{t+i})H^T(\theta_0, z_t))\end{aligned} \tag{2.16}$$

where the value I should be properly selected according to the correlation of the signals. In practice, one can gradually increase the value of I until $\Sigma(\theta_0)$ converges. With these results, detection of small changes in the parameter θ is asymptotically equivalent to the detection of the changes in the mean of a Gaussian vector. The generalized likelihood ratio (GLR) test detecting unknown changes in the mean of a Gaussian vector is a χ^2 test. It is shown that the GLR test of H_1 against H_0 can be written as (Basseville 1998):

$$\begin{aligned}&\chi^2_{global} = \xi_N(\theta_0)^T\Sigma^{-1}(\theta_0)M(\theta_0) \\ &\times(M^T(\theta_0)\Sigma^{-1}(\theta_0)M(\theta_0))^{-1}M^T(\theta_0)\Sigma^{-1}(\theta_0)\xi_N(\theta_0)\end{aligned} \tag{2.17}$$

If $M(\theta_0)$ is a square matrix, then this test can further be simplified to

$$\chi^2_{global} = \xi_N(\theta_0)^T\Sigma^{-1}(\theta_0)\xi_N(\theta_0) \tag{2.18}$$

χ^2_{global} has a central χ^2 distribution under H_0, and a non central χ^2 distribution under H_1. The degree of freedom of χ^2_{global} is the row dimension of θ. A threshold value χ^2_α can be found from a χ^2 table, where α is the false alarm rate specified by the users. If χ^2_{global} is found to be larger than the threshold value, then a change in the parameter is detected.

2.3 Robustness issues

Some of the restrictions and constraints put on the initial formulation of the local approach are that the true system is in the model set, the monitoring algorithm is derived from a bias free identification scheme and the same identification procedure is used to estimate the nominal system parameter and to derive the monitoring scheme (Benveniste *et al.* 1987). Zhang *et al.* (1994) show that this approach has wider applicability than initially stated and can be used for cases where there is model order reduction (model and plant structures are different), cases where true system is not in the model set, cases where biased identification procedures are used. Also they show that instead of the identification simpler Monte-Carlo

estimation technique can be used. Hence the initial constraints and restrictions can be relaxed. A simple understanding of this work by Zhang *et al.* (1994) is given here.

The initial restriction on the local approach is that there should be no structural mismatch between the model and the plant, i.e. as the number of data points tends to infinity the following equation for the primary residual holds:

$$\begin{aligned} E[H(\theta_0, X)] &= 0 \\ \text{or } \frac{1}{N}\sum_{k=1}^{K} H(\theta_0, X_k) &= 0 \\ \text{as } N \to \infty & \end{aligned} \tag{2.19}$$

where Θ is the actual plant, θ_0 is the nominal model ($\theta_0 \to \Theta$, i.e assuming consistent, unbiased identification), X_k is the input-output measurements or the regressor and $k = 1, ..., K$ is the number of measurements. This relation will not hold in the case of some biased identification algorithm. This requirement must be relaxed for wider applicability of the local approach.

To work around this problem for the case that there is a mismatch between the model θ and the plant Θ a function $\mathcal{F}$ is introduced such that:

$$\theta \to \mathcal{F}(\Theta) \qquad \text{as} \qquad N \to \infty \tag{2.20}$$

$\mathcal{F}$ is called the bias function and defines a relationship between the actual system (Θ) and the identified model (θ). This means that as the number of data points tend to infinity the model parameters will approach some functional form of the original system parameters because of the model-plant mismatch. If there is no model-plant mismatch the model parameters will approach the actual system parameters (equation 2.19). Using this notation of the bias function the hypothesis test of parameter change using equation 2.20, 2.1 and 2.2 will be:

$$\begin{aligned} \mathcal{H}_0 &: \quad \mathcal{F}(\Theta) = \theta_0 \\ \mathcal{H}_1 &: \quad \mathcal{F}(\Theta) = \theta_0 + \frac{\eta}{\sqrt{N}} \end{aligned}$$

Here again Θ is the actual parameter, θ_0 is the nominal identified parameter and η is some arbitrary vector which signifies a fixed change in any arbitrary direction. $\mathcal{F}$ is the bias function of the primary residual. Here the model is being checked to see if its functional is still in agreement with a new set of data. From this hypothesis test the detection algorithm can be derived on similar lines as in section 2.2.1 taking into account equation 2.20. Here it is also important to note that the primary residual may not have zero mean due to the model-plant mismatch (equation 2.20) but can be made to have zero mean by a bias correction term:

$$H_0(\theta_0, X_k) \triangleq H(\theta_0, X_k) - h_0$$

where the bias term can be approximated as:

$$h_0 \approx \hat{h}_0 \triangleq \frac{1}{K}\sum_{k=1}^{K} H(\theta_0, X_k)$$

For the detection test the normalized residual is defined as:

$$\xi_N(\theta_0) \triangleq \sum_{k=1}^{N}(H(\theta_0, X_k) - h_0)$$

This normalized residual would have the properties of being zero mean for no change in its functional parameters and non-zero mean otherwise. Hence the detection problem reduces to monitoring of the mean of this normalized residual. Here the changes in the bias function $\mathcal{F}$ are to be detected and the changes in Θ which are not captured by the bias function cannot be detected. So the local approach can still be used for change detection despite the mismatch in the plant and the model, the only drawback being that the changes in the plant not captured by the bias function will not be detected. Hence there will be a loss of information due to the mismatch between the model and the plant. An important thing to note is that knowledge of the bias function $\mathcal{F}$ is not needed for the detection algorithm.

Another contribution of this work is application of the local approach to the cases where it is not possible to arrive at a stable and convergent recursive algorithm for identifying the nominal model θ_0. This method is applicable to cases where a stable convergent recursive algorithm is not possible. In these cases offline arbitrary procedure may be used for identification. It may not be possible to derive the primary and normalized residuals from this arbitrary identification procedure. For this an alternate method which uses Monte-Carlo estimation is proposed (Zhang *et al.* 1994):

1. Choose a nominal model denoted by θ_0 (this may be derived by any method - not necessarily by identification)

2. Calculate the primary residual associated with some least square criterion (equation 2.6). We may not be able to use this criterion for our identification because of its poor convergence behavior. Here the covariance and the bias can be calculated by the Monte-Carlo estimates. the bias would be given by:

$$h_0 \approx \hat{h}_0 \triangleq \frac{1}{K}\sum_{k=1}^{K} H(\theta_0, X_k)$$

3. $H_0(\theta_0, X_k) \triangleq H(\theta_0, X_k) - h_0$ is used as the primary residual for the monitoring which is carried out as in section 2.2.

So this method using Monte-Carlo estimation provides an attractive way to solve the detection problem when the identification scheme is not dependable as the nominal model can in principal be arbitrarily selected. It is also shown that the use of this method may result in a loss of efficiency.

2.4 Dealing with non-linear dynamic systems

This section refers to the work of Zhang *et al.* (1998) and the references therein. In this work the local approach which is proposed for fault detection using input-output representation of the system, is extended to the case of monitoring nonlinear dynamic systems. Recent results in input-output representation of nonlinear system (Ljung and Glad 1994) have been used to this purpose.

2.4.1 Representation of nonlinear systems by differential algebraic equations (DAEs)

Since models for system formed using engineering laws involve differential equations, nonlinear models are often of interest for control and fault detection. An example will be continuous time state space nonlinear dynamic systems which are expressed as:

$$\dot{x} = f(x, u, \theta) \tag{2.21}$$

$$y = g(x, u, \theta) \tag{2.22}$$

where f and g are nonlinear functions; x is the state variable; u and y are the input and output signals; $\dot{x}$ is the time derivatives of x; θ is the model parameters. Systems modeled by Differential algebraic equations (DAEs) are considered, specifically of the form:

$$f_i(x, u, y, \theta, p) = 0, \qquad i = 1, 2, ..., r \tag{2.23}$$

where $f_i's$ are the differential polynomials in x, u, y, θ (polynomial in x, u, y, θ and their time derivatives). p is the derivative operator. Restricting the $f_i's$ to be polynomials makes it possible to use the results in differential algebra and is not a restrictive condition. Many non polynomial nonlinearities can be transformed to equivalent polynomial DAEs. The main problem of applying fault detection and isolation approach to dynamic systems is unknown or unmeasured state variables x. Normally this problem is dealt with by either estimation or elimination. The estimation is finding observers or filters (for stochastic systems). There has been a lot of study on design of observers or filters for linear system and their application in fault detection (Frank 1990) but little work has been done on the nonlinear systems modeled by DAEs. On the other hand the procedure of elimination is finding analytical redundancy in the model. By eliminating the states x the mathematical model is reduced to an equivalent input output representation of the system (like parity check for linear systems). In this work the case of elimination of x in nonlinear systems modeled by DAEs is considered by Zhang and his co-workers. In the approach considered both elimination and estimation are involved - this is because the elimination of x leaves time derivatives of u and y, which is an estimation problem. The use of continuous time models represented by DAEs is motivated by the fact that physical laws are expressed by differential equations and also mathematical tools are available to handle the DAEs.

2.4.2 Elimination of unknown variables

Zhang and co-workers have made use of a general method for checking the identifiability of a system (Ljung and Glad 1994). The by-product of this method is that a input output representation of the system can be obtained. As mentioned before since a large class of nonlinear dynamic system can be represented as DAEs this method promises wide applicability. A summary of the method is described here.

Assume the system is modeled by a set of r DAEs given by equation 2.23 where the functions $f_i's$ are differential polynomials in x, u, y and θ. One can obtain many differential polynomials in the same variable by adding several f_i's differentiating one f_i in time, or multiplying a f_i by any polynomial. By repeatedly performing such operations an infinite number of differential polynomials can be obtained . Such a set of infinite differential polynomials is called the differential ideal formed by the f_i's (denoted by 'F' from now on). Each differential polynomial ϕ in F corresponds to a DAE $\phi = 0$.

If there exists $x(t)$ and $\theta(t)$ such that signals $u(t)$ and $y(t)$ satisfy a set of polynomial DAEs, then that pair of $u(t)$ and $y(t)$ constitutes a solution of the corresponding polynomials. Also it is obvious that since $y(t)$ and $u(t)$ are the solution to the original set of DAEs (2.23) they are also a solution of the differential ideal F. From the infinite set of F we can select of finite subset to specify a solution which satisfies some criterion. For this case the subset of interest is that subset of F which does not involve the variable x. In the work of Ljung and Glad (1994), Ritt's algorithm (Ritt 1950) is used as a tool to find characteristics sets with desirable properties from a differential ideal (can be visualized as Gramm-Schmidt or Gauss elimination schemes in linear algebra).

The theorem of global identifiability (Ljung and Glad 1994) can be stated as follows. If there exists a characteristic set of F including differential polynomials of the form

$$P_j(u, y, p)\theta_{(j)} - Q_j(u, y, p) \tag{2.24}$$

for all $j = 1, 2, ...n$ (n is the dimension of θ and θ_j is the j_{th} component), where $P_j(u, y, p)$ and $Q_j(u, y, p)$ are differential polynomials in u and y, then θ is globally identifiable, i.e.θ can be uniquely determined in C^n under the condition of persistent excitation. Another aspect of equation 2.24 is that the parameter components of θ are decoupled along with the elimination of x, but this is at a cost of high order derivative terms in u and y. So a solution which gives an input output representation involving low order derivatives is prefered for practical purposes even if it may result in a less decoupled θ. So within F a differential polynomial of the form:

$$g_i(u, y, \theta, p) \qquad\qquad i = 1, 2, ..$$

is searched where x is eliminated but the components of θ may be involved in each polynomial $g_i(u, y, \theta, p)$. Ritt's algorithm can be used to search in F for a particular $g(u, y, \theta, p)$

(dropping the subscript i for compactness) which is linear in θ or in other words of the form:

$$g(u, y, \theta, p) = P(u, y, p)\theta - Q(u, y, p) \tag{2.25}$$

where $P(u, y, p)$ and $Q(u, y, p)$ are polynomial vectors or matrices. The DAE corresponding to this will be:

$$g(u, y, \theta, p) = 0 \tag{2.26}$$

or using equation 2.25

$$P(u, y, p)\theta - Q(u, y, p) = 0 \tag{2.27}$$

which is the input output representation of the system. The condition for equation 2.25 to exist in F is that θ should be globally identifiable (Ljung and Glad 1994). If θ is not globally identifiable but locally identifiable then a differential polynomial nonlinear in θ is found which will still give the input output representation of the system for the purpose of fault detection and isolation.

2.4.3 Residual evaluation for fault detection

The results discussed so far have been limited to deterministic models, but they have also been extended to the stochastic case to take into account modeling uncertainty and measurement error by Zhang and co-workers (1998). For nonlinear systems it is not realistic to make the general assumptions as made in linear system that the noises are additive in both state and observation equation. Usually noise models are assumed based on first principal knowledge of the system. For simplicity the authors have assumed the following formulation:

$$g(u_k, y_k, \theta, \delta) = \epsilon_k \tag{2.28}$$

(the original equation has been discretized such that $u_k = u(k\tau)$ and $y_k = y(k\tau)$ with $k = 1, 2, ..., N$. δ is the discrete approximation of the differentiation operator p). The local approach for fault detection and isolation deals with finding a vector valued quantity called the primary residual from the input output representation of the system to be monitored (section 2.2). This primary residuals should be zero mean for no fault and non-zero mean otherwise. Normally, if θ is obtained by minimizing some criterion then the gradient of this criterion w.r.t θ is taken as the primary residual. Consider the system modeled by equation 2.28 and assume that it is locally identifiable (Ljung and Glad 1994) at the nominal value $\theta = \theta_0$, also assume that the component polynomial in equation 2.28 has at least one term which is independent of θ. From these assumptions the primary residual can be formed, it will be related to the identification of θ by minimizing the square of $g(u_k, y_k, \theta, \delta)$:

$$\begin{aligned} H(u_k, y_k, \theta, \delta) &= \frac{1}{2}\frac{\delta}{\delta\theta}(g^T(u_k, y_k, \theta, \delta)g(u_k, y_k, \theta, \delta)) \\ &= (\frac{\delta g(u_k, y_k, \theta, \delta)}{\delta\theta})^T g(u_k, y_k, \theta, \delta) \end{aligned} \tag{2.29}$$

The assumption here is that for this vector $H(u_k, y_k, \theta, \delta)$ equation 2.9 and 2.10 should be satisfied. This may not be a realistic assumption for nonlinear systems but are taken for simplicity of the presentation. Unfortunately even if e_k in equation 2.28 is Gaussian white noise not much can be said about the distribution of the primary residual. For this purpose the authors of this work used the normalized residual based on the assumption that the primary residual $H(u_k, y_k, \theta, \delta)$ is differentiable in θ and the following quantities exist:

$$M(\theta_0) = \mathbf{E}_{\theta_0}(\frac{\delta}{\delta\theta} H(u_k, y_k, \theta, \delta) \mid_{\theta=\theta_0}) \tag{2.30}$$

$$\Sigma(\theta_0) = \sum_{k=-\infty}^{+\infty} \mathrm{cov}_{\theta_0}(H(u_k, y_k, \theta, \delta), H(u_k, y_k, \theta, \delta)) \tag{2.31}$$

where $\mathbf{E}_{\theta_0}$ and cov_{θ_0} are respectively the expectation and the covariance when the system is at its nominal state. Here it is also assumed that $M(\theta_0)$ is full column rank and $\Sigma(\theta_0)$ is positive definite. Taking these assumptions, equation 2.9 and 2.10 and the hypothesis tests for a normalized residual is defined:

$$\xi_N(\theta_0) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} H(u_k, y_k, \theta, \delta) \tag{2.32}$$

which converges weakly to a random vector with a Gaussian distribution when N goes to infinity. Taking a specific example where:

$$g(u_k, y_k, \theta, \delta) = P(u_k, y_k, \delta)\theta - Q(u_k, y_k, \delta) \tag{2.33}$$

the corresponding stochastic model being:

$$P(u_k, y_k, \delta)\theta - Q(u_k, y_k, \delta) = \epsilon_k \tag{2.34}$$

then the primary residual writes as:

$$H(u_k, y_k, \delta) = (P^T(u_k, y_k, \delta)P(u_k, y_k, \delta))\theta - (P^T(u_k, y_k, \delta)Q(u_k, y_k, \delta))$$

and

$$M(\theta_0) = \mathbf{E}_{\theta_0}(P^T(u_k, y_k, \delta)P(u_k, y_k, \delta))$$

which is invertible if the system is persistently excited. From equation 2.34 we have

$$H(u_k, y_k, \delta) = P^T(u_k, y_k, \delta)\epsilon_k$$

and

$$\Sigma(\theta_0) = \sum_{k=-\infty}^{+\infty} \mathrm{cov}_{\theta_0}(P(u_1, y_1, \delta)\epsilon_1, P(u_k, y_k, \delta)\epsilon_k)$$

and if ϵ_k is assumed to be an independent sequence with variance Σ_ϵ then

$$\Sigma(\theta_0) = \mathbf{E}_{\theta_0}(P(u_k, y_k, \delta)\Sigma_\epsilon P(u_k, y_k, \delta))$$

From this point process monitoring can be done on the lines described in section 2.2.

To demonstrate the local approach application for nonlinear dynamic system a microbial growth process with continously fed substrate is considered by the authors (Zhang *et al.* 1998). The Monad model (Monad 1950, Abhorey and Williamson 1978) given by the following equations is used:

$$\begin{aligned} \dot{x}_1 &= \frac{a x_1 x_2}{x_2 + b} - x_1 u \\ \dot{x}_2 &= \frac{da x_1 x_2}{x_2 + b} + (c - x_2)u \end{aligned} \tag{2.35}$$

where

x_1: microbial concentration in the process,

x_2: substrate concentration in the process,

u: dilution rate,

a: maximum growth rate,

b: saturation parameter,

c: inlet substrate concentration,

d: yield factor.

u is the input here and x_1, x_2 are the state variables, a, b, c, d are parameters and x_2 is assumed to be observed, i.e:

$$y = x_2$$

This model is first transformed into a polynomial DAE:

$$(x_2 + b)\dot{x}_1 - a x_1 x_2 + (x_2 + b)x_1 u = 0$$

$$(x_2 + b)\dot{x}_2 - da x_1 x_2 - (x_2 + b)(c - x_2)u = 0 \tag{2.36}$$

$$y - x_2 = 0 \tag{2.37}$$

(parameter d can neither be identified or monitored here using the inputs and outputs) From this it is shown that an input output representation of the system can be obtained by applying Ritt's algorithm (Ljung and Glad 1994). The input output representation is given by:

$$\begin{aligned} acuy^2 - ay^2\dot{y} &- auy^3 + bcu\dot{y} - bc\dot{u}y - \\ bcu^2 y &+ b\dot{u}y^2 - b\dot{y}^2 + bu^2y^2 + buy\dot{y} - cu^2y^2 - c\dot{u}y^2 + \dot{u}y^3 \\ +y^2\ddot{y} &- cu^2y^2 - c\dot{u}y^2 + \dot{u}y^3 + y^2\ddot{y} + u^2y^3 + 2uy^2\dot{y} = 0 \end{aligned} \tag{2.38}$$

Here it can be seen that the non identifiable parameter d does not figure in the input-output representation of the system. This equation can be denoted in a discretized form as:

$$g(u_k, y_k, \theta, \delta) = \epsilon_k \tag{2.39}$$

where ϵ_k is the modeling error. From this result the implementation becomes pretty straight forward. The time derivatives are estimated by numerical methods. In Zhang (1998) the data is prefiltered by a low pass filter, to get better estimates of the derivatives. The primary and normalized residuals can be obtained from equation 2.39 and the local approach monitoring algorithm can be applied. Numerical evaluations as shown in the work (Zhang *et al.* 1998), show that the approach is effective for detecting faults in dynamic non-linear systems.

2.5 Isolation

Under the framework of the local approach methods for isolation of the faulty parameter has also been proposed (Zhang *et al.* 1998, Basseville and Nikiforov 1993). The results from these references have been presented in this section.

Once the decision has been made that the data represents a faulty mode of the system model being monitored it is important to isolate the fault. This means to isolate the particular part of the parameter vector being monitored which is faulty. Representing this as a hypothesis test the problem is to decide between the hypothesis (for all sub vectors η_a of η):

$$\mathcal{H}_0(\eta_a) : \eta_a = 0$$

and

$$\mathcal{H}_1(\eta_a) : \eta_a \neq 0$$

where η_a can be taken as the first component of η. An iterative search using all the permutations can be performed. Let η_b be the complimentary vector of η_a such that:

$$n = \begin{bmatrix} \eta_a \\ \eta_b \end{bmatrix} \tag{2.40}$$

n_a and n_b are the dimensions of η_a and η_b respectively. The isolation problem is dealt with in two ways. The first is assuming η_b to be zero (called sensitivity test) and the second is to assume it to be the nuisance in order to make the decision between the hypotheses (called min-max test).

2.5.1 Sensitivity test

In this approach the matrix M is partitioned corresponding to the partitioning given by equation 2.40:

$$M = [M_a \quad M_b]$$

such that $M\eta = M_a\eta_a + M_b\eta_b$. The sensitivity test which assumes that $\eta_b = 0$ is a χ^2 test with n_a degrees of freedom given by:

$$\tilde{\chi}_a^2 = \tilde{\xi}_a^T F_{aa}^{-1} \tilde{\xi}_a$$

where

$$\tilde{\xi}_a = M_a^T \Sigma^{-1} \xi$$

is the partial score in η_a, and

$$F_{aa} = M_a^T \Sigma^{-1} M_a$$

is the covariance matrix of $\tilde{\xi}_a$

2.5.2 Minmax test

In the minmax test η_b is considered to be non zero but is considered as the nuisance parameter and is statistically rejected. Here the matrix F is given by:

$$F = M^T \Sigma^{-1} M$$

is partitioned as

$$F = \begin{bmatrix} F_{aa} & F_{ab} \\ F_{ba} & F_{bb} \end{bmatrix} = \begin{bmatrix} M_a^T \Sigma^{-1} M_a & M_a^T \Sigma^{-1} M_b \\ M_b^T \Sigma^{-1} M_a & M_b^T \Sigma^{-1} M_b \end{bmatrix}$$

The residual of the linear regression of the partial score $\tilde{\xi}_a = M_a^T \Sigma^{-1} \xi$ with respect to the partial score $\tilde{\xi}_b = M_b^T \Sigma^{-1} \xi$ is considered:

$$\xi_a^* = \tilde{\xi}_a - F_{ab} F_{bb}^{-1} \tilde{\xi}_b$$

then the minmax test can be stated as a χ^2 test with n_a degrees of freedom:

$$\chi_a^{2*} = \xi_a^{*T} F_a^{*-1} \xi_a^*$$

where

$$F_a^* = F_{aa} - F_{ab} F_{bb}^{-1} F_{ba}$$

is the covariance matrix of ξ_a^*. Both these tests can be implemented as repetitive schemes to locate the faulty parameter in vector θ. The challenge lies in setting the threshold value for both the $\tilde{\chi}_a^2$ and χ^{2*} tests involved in sensitivity and minmax tests. This is because though ideally both the test should follow a central χ^2 distribution when $n_a = 0$, it is not so in practice because of approximation in evaluation of ξ. A practical solution proposed in Zhang *et al.* (1994) is to compute the minmax test for each possible sub-vector of η_a. Then the subvector corresponding to the largest test value will be the non zero subvector (hence the subvector containing the faulty parameters).

Table 2.1: *Results of local approach detection algorithm on simulation runs*

Process parameters	Disturbance parameters	mean χ^2	False alarms
no change	no change	8.0499	1
10 % change	no change	13.9557	1
no change	10 % change	37.8760	2
20 % change	no change	8017.8	0
no change	20 % change	60.43	0

2.6 Simulation examples

Example 1.1

The system considered is a simple first order open loop system as shown:

$$y_t = \frac{0.33z^{-4}}{z - 0.73}u_t + \frac{z - 0.4}{z - 0.95}e_t \tag{2.41}$$

Ten percent change in parameters of both process and noise models is made to see the effectiveness of the local approach detection algorithm. The threshold for five percent false alarm rate using the χ^2 tables is 9.49. The results are shown in table 2.1. The input in this case is taken from MATLAB idinput function and the noise as a random number with variance 0.1 and mean zero. No model plant mismatch is assumed and the model structure used for monitoring is the same as the true process model structure.

This simulation shows the power and the sensitivity of the local approach based fault monitoring algorithm. The algorithm is able to detect small changes in the parameters of the process and noise models. In the next example we shall consider the case of under parameterization.

Example 1.2

Another example is considered to illustrate the effect of under parameterization and cases when the model is not in the true plant set (refer to section 2.3). This example is similar to the one considered in Zhang *et al.* (1994) but has a different emphasis. Here a tenth order auto regressive (AR) process is taken and is monitored using AR models of order two, three and four. The process can be represented by:

$$y_k = \sum_{k-i}^{10} a_i y_{k-i} + v_k$$

where a_i are the autoregressive parameters, y_k is the output and v_k is an i.i.d . zero mean and Gaussian distributed noise with variance of 0.1. The process parameter vector is denoted by Θ_0 and we consider two faulty modes of this vector given by Θ_1 and Θ_2. The

Table 2.2: *AR process parameters chosen for the simulation with the value of the parameters for two faulty nodes*

	Θ_0	Θ_1	Θ_2	AR(9)
a_1	-0.2790	-0.3467	-0.2611	-0.3083
a_2	-0.1991	-0.0705	-0.3248	-0.0932
a_3	0.0820	0.0172	0.4674	0.0885
a_4	0.3503	0.3954	0.0754	0.3049
a_5	-0.3312	-0.3504	-0.3420	-0.2599
a_6	0.2850	0.2951	0.2545	0.2165
a_7	-0.0406	-0.0452	0.0368	-0.0655
a_8	-0.0460	-0.0228	-0.0734	-0.4412
a_9	0.0308	0.0152	0.0350	0.1162
a_{10}	0.2309	0.2309	0.2309	-

values for these parameter vectors are kept the same as in Zhang *et al.* (1994) (refer to table 2.2).

The process has been so selected by Zhang and co-workers such that it is not possible to represent it with any AR model with order less than ten. Also the change from Θ_0 to Θ_1 is such that the variance of the output is the same for both the cases and the change from Θ_0 to Θ_2 is such that the first three auto correlation coefficients of the outputs are the same in both the cases. The noise variance is kept fixed at 0.2 throughout. The spectra plots of the various parameterization vectors given above and also a ninth order AR model are given in figure 2.1. The monitoring algorithm is based on three different models given in table 2.3. These three models, each identified using least square, are of the order two, three and four. Spectra plots of the three monitoring models and the actual systems are given in figure 2.2. Each test run is conducted ten times with different driving noise sequence.

The result of running the local approach based detection algorithm on Θ_0 - no change in parameters is shown in table 2.4. As expected all the calculated χ^2 values are below the threshold values (for five percent false alarm rate). The result of running the local approach based detection algorithm on the first faulty mode Θ_1 is shown in table 2.5. Here it can be seen that monitoring algorithm using any of the three monitoring models is able to detect the change with all the calculated χ^2 values above the threshold. Also there are no false alarms. Table 2.6 shows the result of test on the second faulty node Θ_2. It can be seen that while the monitoring algorithm using third and fourth order models detect the change easily, the detection algorithm using the second order model has many false alarms and has a mean calculated value close to the threshold. The loss of detection or poor detection by the second order model can be explained by loss of information due to underparametrization. Although the bias function is not known but it is assumed that some relationship exists between the model identified and the actual system and it is desirable

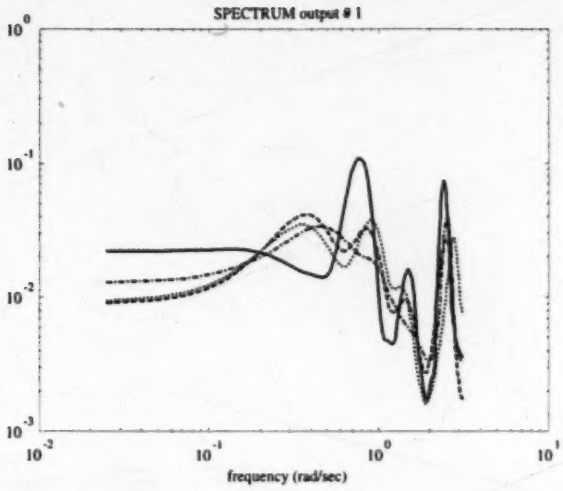

Figure 2.1: spectra plot of the AR processes given in table 2.2, the solid line corresponds to Θ_0, the dashed line Θ_1, the dotted line Θ_2 and the dash dotted line is the least square identified AR(9) model.

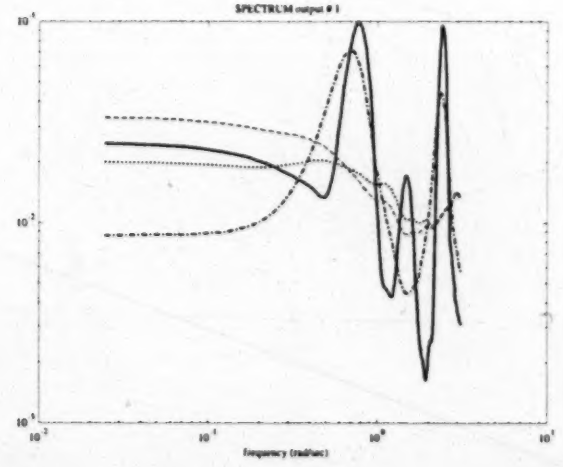

Figure 2.2: spectra plot of the monitoring models and the actual process, the solid line corresponds to actual process, the dashed line the AR(2) model (autoregressive model of order of two), the dotted line the AR(3) model and the dash dotted line is the AR(4) model.

Table 2.3: *AR model parameters chosen for monitoring*

	θ_2	θ_3	θ_4
a_1	-0.1712	-0.1261	-0.1003
a_2	-0.2022	-0.1072	-0.1665
a_3	–	0.0478	-0.0212
a_4	–	–	0.543

Table 2.4: *Calculated χ^2 values for different monitoring θ's with no change Θ_0*

	θ_2	θ_3	θ_4
1	2.6	7.9	7.9
2	0.6	7.9	7.9
3	5.7	7.9	7.9
4	3.5	7.9	7.9
5	0.8	7.9	7.9
6	4.3	7.9	7.9
7	1.2	7.9	7.9
8	5.0	7.9	7.9
9	2.3	7.9	7.9
10	4.6	7.9	7.9
mean	3.8	7.9	7.9
threshold	5.99	7.81	9.49

that it captures all the changes of interest in the actual system to aid in timely detection. This simulation shows that the low order model may not always capture all the changes of interest in the actual system. This is shown by the false alarms in the detection of the second faulty mode in this case, for example.

2.7 Industrial applications

2.7.1 Process monitoring

Process Monitoring is continuous diagnosis of a process for abnormalities and malfunctions. For the industrial case study process monitoring algorithms are implemented on data from a distillation column (Figure 2.3) and a fluidized Coker unit (Figure 2.5) at Syncrude upgrading (Ft. Mcmurray, Canada). The results are compared to the result obtained from a monitoring approach based on singular value decomposition, a brief overview of which is given next. The results for the distillation column are also compared to a PCA based approach.

Table 2.5: *Calculated χ^2 values for different monitoring θ's with change from Θ_0 to Θ_1*

	θ_2	θ_3	θ_4
1	71.1	118.2	253.6
2	23.4	127.9	83.1
3	166.8	185.0	210.6
4	54.6	100.9	306.5
5	38.5	327.9	172.4
6	82.9	149.3	536.7
7	20.5	172.7	258.7
8	111.5	61.9	158.8
9	45.1	190.7	95.6
10	71.2	131.8	200.7
mean	68.5307	156.6	227.6
threshold	5.99	7.81	9.49

Table 2.6: *Calculated χ^2 values for different monitoring θ's with change from Θ_0 to Θ_2*

	θ_2	θ_3	θ_4
1	13.8	205.0	4051.2
2	20.9	380.3	27049.3
3	16.8	193.8	6307.5
4	20.3	379.7	23249.2
5	60.1	2017.1	3345.2
6	42.7	211.5	3976.2
7	4.0	311.4	3154.8
8	26.5	272.4	2626.2
9	5.8	263.4	3409.8
10	10.1	1057.1	5791.0
mean	22.1	529.2	829.59
threshold	5.99	7.81	9.49

2.7.2 Process monitoring based on singular value decomposition

Process monitoring based on the singular value decomposition is a simple yet effective method for detection of process faults proposed by Maryak *al* (1997) . This method of monitoring has been summarized in the section.

For a data matrix A the eigenvalue - eigenvector relation is given by the equation $Ax = \lambda x$, where λ is the eigenvalue and x the corresponding eigenvector. Then for a null eigenvalue (eigenvalue equal to zero) this equation will become $Ax = 0$ (Klema and Laub 1980). The eigenvector corresponding to this null eigenvalue can be said to define the relationship between the various columns of the data matrix A. For this method of monitoring the process variables which seem to be correlated (the exact correlation may not be known) are put in a matrix as column vectors corresponding to the data matrix A. Then over the nominal training data, the null eigenvalue and the corresponding eigenvector are found. In principle the eigenvector should give the same relationship on the test data $Ax = 0$ if no fault has taken place. If the relationship does not hold a possible fault is indicated. For the actual training calculations of the SVD monitoring methodology, correlated data is grouped as column vectors in a matrix, such as:

$$X_m = [x_1, \ldots, x_m]^T \tag{2.42}$$

where $x_1, \ldots, x_m$ are the process variables, or some feature of the data variable. A feature is defined as some transformation of the raw data, such as maximum or minimum value, which summarizes some important aspect of the data and hence can be used as a concise representation of the actual data. This makes handling of large data easier and hence prevents the algorithm from being overwhelmed by data. The 'relationship' vector from the null space of X (the eigenvector corresponding to null eigenvalue) is calculated, say r. In some cases there might be more than one null eigenvalues and consequently more than one corresponding eigenvectors. In these cases there will be a matrix with all such eigenvectors set as column vectors such as:

$$R_m = [r_m, s_m, t_m, \ldots] \tag{2.43}$$

where $r_m, s_m, t_m, \ldots$ etc. are orthogonal, eigenvectors which correspond to the more than one null eigenvalues (eigenvalues which have zero value or are less than one percent of the maximum eigenvalue). The fault detection test is done by taking the average of the test data of say n sampling instances and then computing the value of the test statistic:

$$\bar{y}_n \hat{T_m} \tag{2.44}$$

where:

$$\bar{y}_n = \frac{y_1 + \ldots + y_n}{n} \tag{2.45}$$

for n sampling instances, and

$$\hat{T_m} = \hat{r}_m + \hat{s}_m + \hat{t}_m + \dots \tag{2.46}$$

the sum of the eigenvectors corresponding to the null eigenvalue. Due to the dynamics of the data matrix being monitored the test statistic (equation 2.44) may never be equal to zero, even for a condition of no fault. Hence a threshold is calculated with five percent allowable false alarm rate corresponding to ninety five percent confidence interval (see Maryak *et al.* 1997, for details for deriving this threshold). The threshold value is computed as:

$$Threshold = 1.96\sqrt{(\frac{1}{m} + \frac{1}{n})\hat{T}_m^T V \hat{T}_m} \tag{2.47}$$

where n is the number of test data points over which the average is taken on the test data, m is the number of samples over which the training of the algorithm is done and V is the variance of random component of the measurements. The value $\hat{T}_m^T V \hat{T}_m$can be estimated by the sample variance of $\hat{T}_m^T x_1, \dots, \hat{T}_m^T x_m$. A possible fault is indicated if the computed value of the test statistic is more than the threshold value.

Comparison of SVD based approach with PCA based approach

An important point to mention is difference between the SVD approach for monitoring and the PCA approach (Huang 1999*b*) . While both approaches use singular value decomposition the PCA approach uses the first few principal Components which together characterize most of the variance of the system. So a PCA based monitoring scheme will check to see if the same first few principal components identified over the nominal training can be used to describe the test data. If the test data can still be described by the principal components identified over the nominal training data it can be said that no fault has taken place. The rest of the eigenvectors especially the ones corresponding to null eigenvalues are discarded. In contrast the SVD approach uses only the eigenvectors corresponding to the null eigenvalues and discards the other eigenvalues and eigenvectors. As already explained in section 2.7.2 this approach uses the property that the last principal component (eigenvector corresponding to the null eigenvalue) describes a relationship between the different variables. It is this relationship which is monitored over the test data. If the last principal component identified over the nominal training data can still be used to describe the relationship in the testing data then it can be said that no fault has occured.

Implementation of SVD based monitoring approach to industrial data

For this particular implementation all the controlled variables, manipulated variables and the disturbance variables of the distillation column (Figure 2.3) are taken in a matrix as column vectors. The variables that do not show correlation are dropped (correlation determined from historized data of all the variables). The SVD monitoring algorithm is

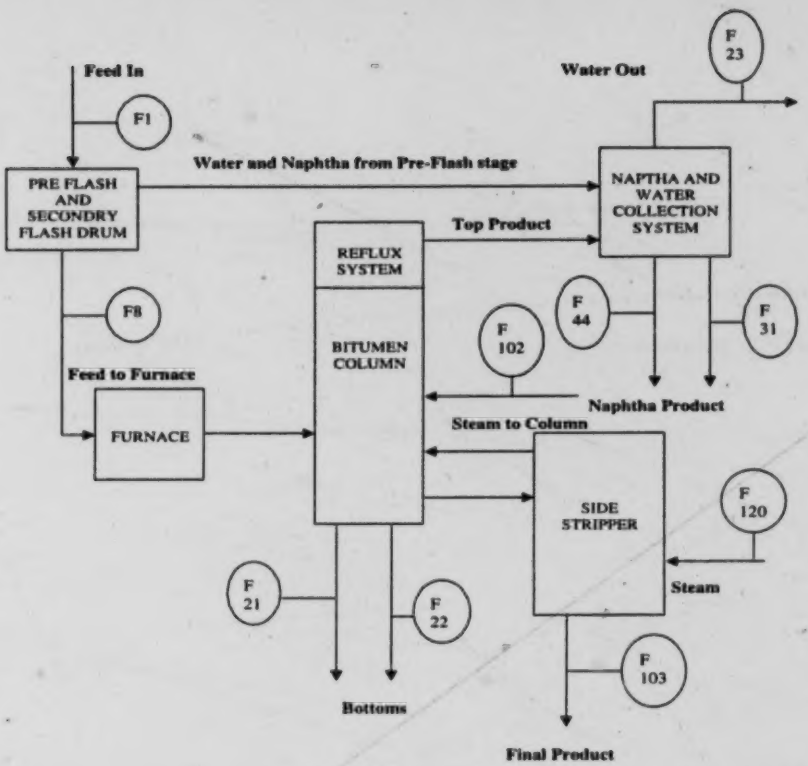

Figure 2.3: The simplified flow diagram of the industrial distillation column used for the mass balance analysis showing the flow meters.

run over a known faulty node (historized flooding data) to see if it can possibly be used as an early warning indicator. From figure 2.4 it can be seen that while the top tray temperature [1](currently used by the operators as a dependable visual indicator of flooding) showed change after around two thousandth sampling instance the SVD monitoring scheme showed a fault around fifty to hundred sampling instances before that. The results improve with a delay structure taken into account as far as the early detection is concerned. Another similar implementation is tried on the Fluidized Coker unit (figure 2.5).

The data matrix in this case is composed of eight seemingly correlated controlled variables, manipulated variables and disturbance variables. The results on a faulty node are shown in figure 2.6.

Figure 2.6 shows that the algorithm is able to detect the fault but is not impressive as far as early detection is concerned. Possible improvements like taking delay structure into

[1]the industrial data in this chapter is not drawn to scale in order to plot the variables shown in the same figure

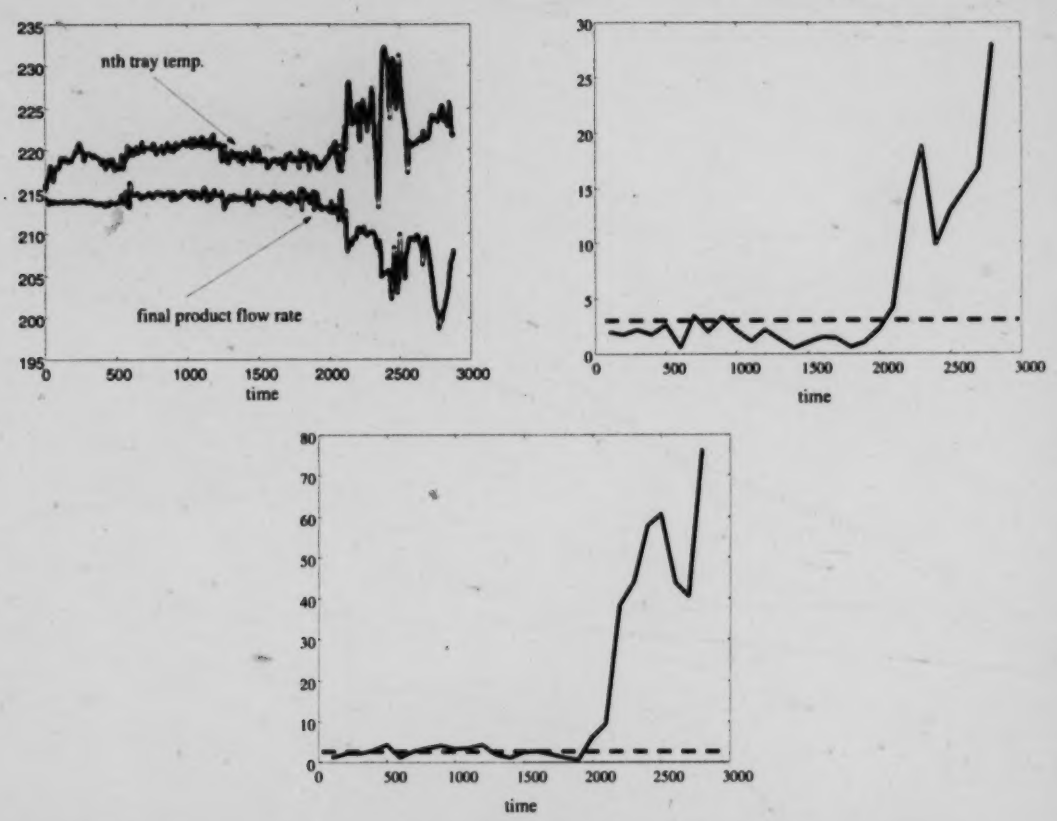

Figure 2.4: (top-left) two variables representative of the flooding (top-right) SVD monitoring plot without the delay being taken into account (bottom) SVD monitoring plot with some approximate delay structure being taken into account - the dotted line shows the threshold.

account or taking more variables into the monitoring matrix can be investigated. For this analysis certain variables are ignored as dependable measurements can not be obtained. The inclusion of these variables can be experimented with to see if it can possibly improve results.

2.7.3 Monitoring Based on local approach and least squares modeling

Basic principle

This approach for process monitoring combines the least squares approach for system modeling with the local approach. For this monitoring algorithm a steady state mass balance equation is formulated:

$$\text{Mass In} - \text{Mass Out} = 0 \tag{2.48}$$

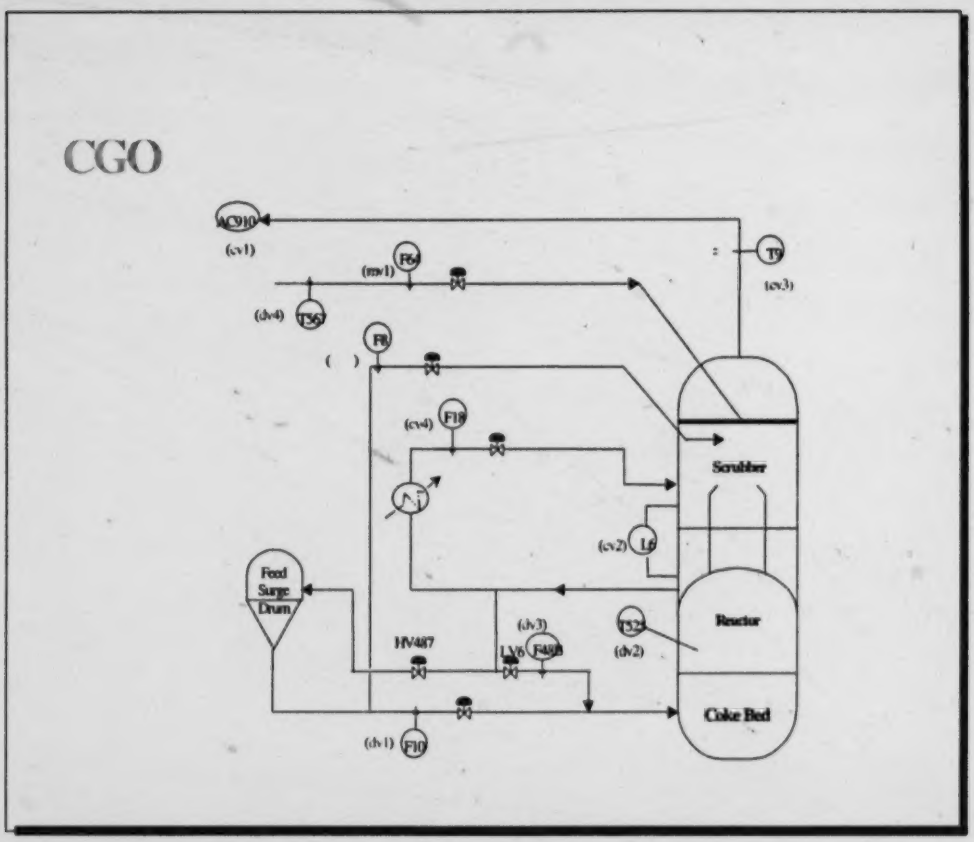

Figure 2.5: Industrial Fluidized Coker with all controlled variables, manipulated variables, disturbance variables.

This steady state mass balance equation is used as a regression model:

$$aA = bB + cC + ...zZ \tag{2.49}$$

The parameters $(a, b, \ldots, z)$ of this regression equation are estimated by method of least squares. These parameters are then monitored by using local approach.
Consider that the variables in a process follow a linear regression model (Huang 1999*a*) (for example in this case study the regression model is formed from the mass balance equation)

$$z_1(n) = \beta_2 z_2(n) + \beta_3 z_3(n) + \ldots + \beta_p z_p(n) + r(n) \tag{2.50}$$

Define a parameter vector:

$$\theta = \begin{bmatrix} \beta_2 \\ \beta_3 \\ \ldots \\ \ldots \\ \beta_p \end{bmatrix} \tag{2.51}$$

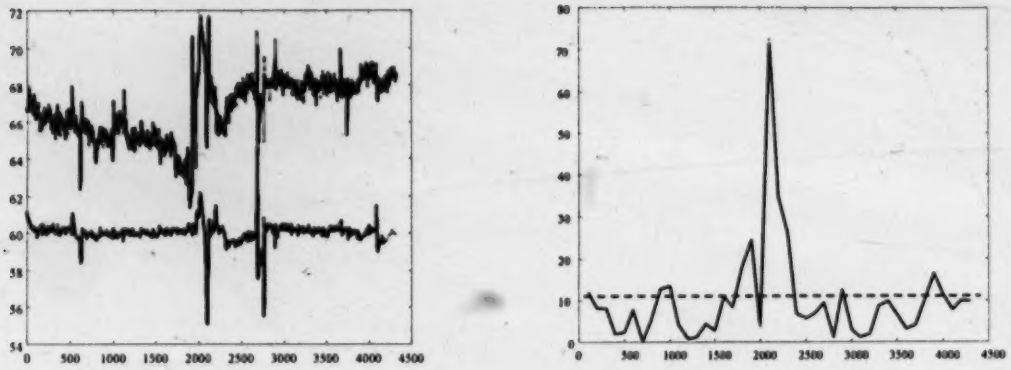

Figure 2.6: (top) two representative variables of a possible faulty node in the Fluidized Coker and (bottom) the resulting SVD monitoring plot - the dotted line shows the threshold.

and a regressor vector:

$$\phi_n = \begin{bmatrix} z_2(n) \\ z_3(n) \\ \dots \\ \dots \\ z_p(n) \end{bmatrix} \tag{2.52}$$

Then the linear regression model can be concisely written as:

$$z_1(n) = \phi_n^T \theta \tag{2.53}$$

The objective function of the least squares (which has to be minimized to evaluate the parameter) function is (for time n = 1 to N):

$$J = \sum_{n=1}^{N} (z_1(n) - \beta_2 z_2(n) - \ldots - \beta_p z_p(n))^2 \tag{2.54}$$

$$= \sum_{n=1}^{N} (z_1(n) - \phi_n^T \theta)^2 \tag{2.55}$$

If the parameter that minimizes the objective function is denoted by θ_0. It can be calculated by solving:

$$\frac{\partial J}{\partial \theta} = -2 \sum_{n=1}^{N} \phi_n (z_1(n) - \phi_n^T \theta) = 0 \tag{2.56}$$

$$\Rightarrow \sum_{n=1}^{N} \phi_n (z_1(n) - \phi_n^T \theta_0) = 0 \tag{2.57}$$

$$\Rightarrow \theta_0 = (\sum_{n=1}^{N} \phi_n \phi_n^T)^{-1} \sum_{n=1}^{N} \phi_n z_1(n) \tag{2.58}$$

which is the least square solution, It has been shown in Huang (1999a) that this also coincides with the eigenvector corresponding to the null eigenvalue by rearranging the variables. The null eigenvalue is given by:

$$\lambda_0 = \frac{1}{N} l_0^T \sum_{n=1}^{N} Z_n Z_n^T l_0 \tag{2.59}$$

where

$$\begin{aligned} l_0 &= [1; \theta_0] \\ Z_n &= [z_1(n), z_2(n), \ldots, z_p(n)] \end{aligned} \tag{2.60}$$

where $z_1(n), \ldots, z_p(n)$ are column vectors corresponding to the time series of various variables. The primary residual in this case is defined by:

$$H(Z_n, l_0) = Z_n Z_n^T l_0 - \lambda_0 l_0 \tag{2.61}$$

The normalized residual is given by:

$$\xi(l_0) = \frac{1}{\sqrt{N}} [\sum_{n=1}^{N} (Z_n Z_n^T) - N \lambda_0 l_0] \tag{2.62}$$

The mean gradient is given by:

$$M(l_0) = \frac{\partial}{\partial l} [\frac{1}{N} \sum_{n=1}^{N} (Z_n Z_n^T l_0) - \lambda_0 l_0] \mid_{l=l_0} \tag{2.63}$$

The covariance matrix is calculated as before using equation 2.16 (for detailed derivation of these relations refer Huang (1999a)). Using these relations for normalized residual and covariance matrix the monitoring problem can be dealt by the local approach.

The idea here is to aim for better monitoring by arriving at a regression equation which should hold for steady state operation (i.e. the steady state mass balance equation) and then to monitor this equation or more correctly the parameter vector of this equation over the test data.

Results

This algorithm is implemented on the industrial distillation column with flooding to see if it can be used for early detection of flooding and if this scheme provides an improvement over the earlier method of monitoring which used singular value decomposition.

It can be seen from figure 2.7 that detection is more than two to three hundred sampling instances before the flooding becomes visible to the operator (who monitors top tray temperature), hence it is worthwhile to fine tune this algorithm to supplement the existing flooding variable. Also false alarms can be seen to have been reduced by the incorporation

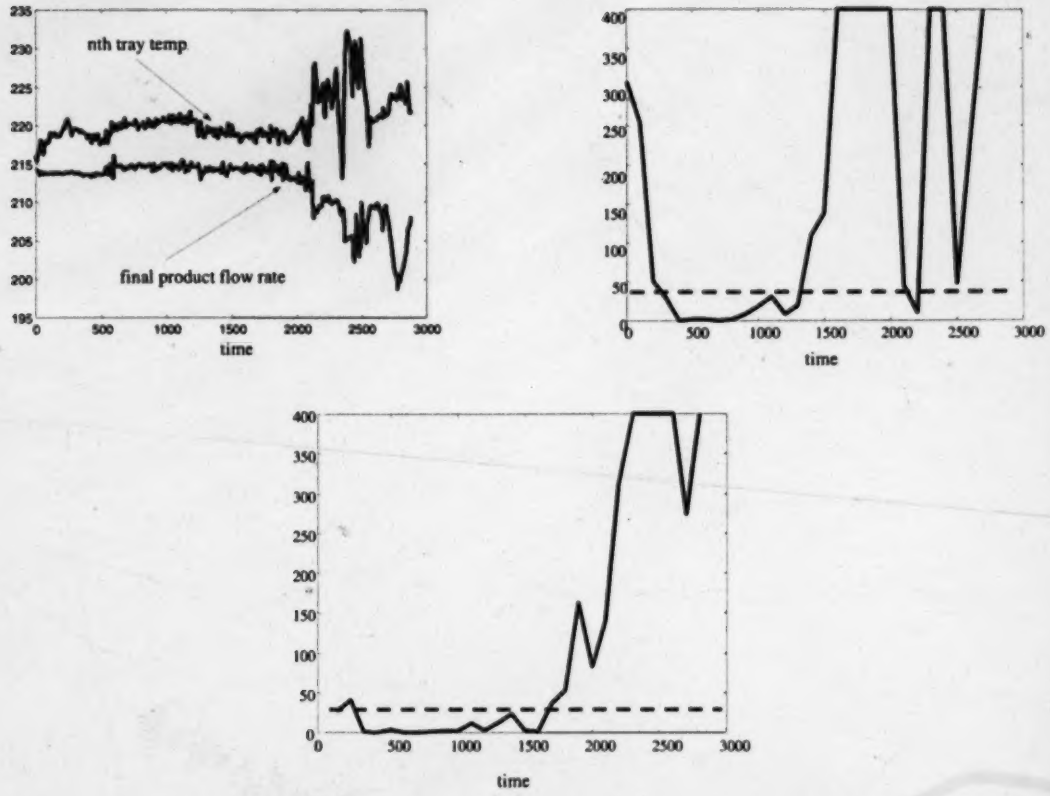

Figure 2.7: (top-left) two variables representative of the flooding (top-right) χ^2 plot without the delay being taken into account (bottom) χ^2 plot with some approximate delay structure being taken into account - the dotted line shows the threshold.

of approximate delay structures. Hence these results show a considerable improvement over the singular value decomposition based monitoring technique.

The monitoring done over another known faulty node of the industrial distillation column is shown in figure 2.8. Here again the results indicate flooding around two to three hundred sampling instances before the visual indication. The results on the two different flooding data show promising results for the LS/Local approach monitoring algorithm in early detection of flooding in the industrial distillation column. Also false alarms are reduced when approximate delay structures are included in the variables. The delay structures taken are purely intuitive. More realistic delay structures can give better results. The algorithm shows versatility as far as reproducibility of results is concerned (successfully tried over many data sets).

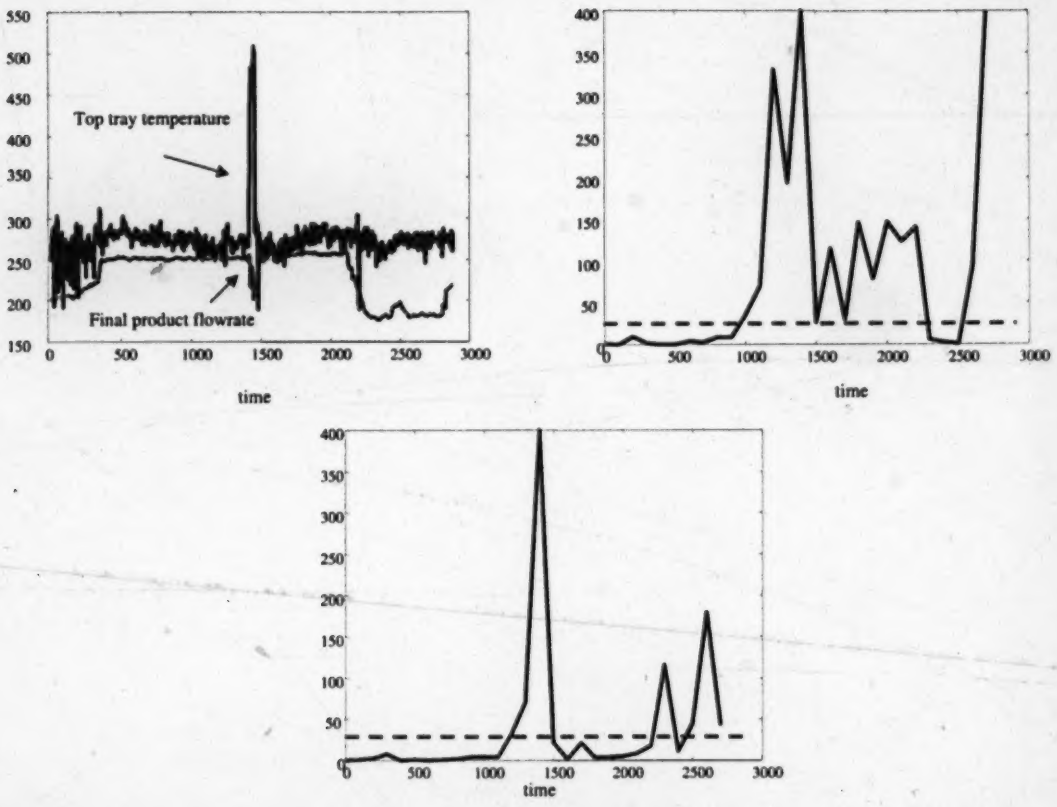

Figure 2.8: (top-left) two variables representative of the flooding (top-right) χ^2 plot without the delay being taken into account (bottom) χ^2 plot with some approximate delay structure being taken into account - the dotted line shows the threshold.

2.7.4 Comparison of the local approach with PCA

The flooding data given in figure 2.7 is also analyzed using principal component analysis. Without going into the details of this approach a commercial software *ProcessDoc* is used to generate the monitoring plots. Readers are referred to PhD thesis of Lakshminarayan (1997) for detailed discussion on PCA/PLS. For this analysis the same data as used for the local approach least square based analysis is taken. Also the same approximate delay structures are taken into account as for the local approach and SVD analysis of the data. These data are correlated due to mass balance law. Figure 2.9 is the squared prediction error plot got from PCA. The first thing that is striking here is that the false alarm rate is higher than the local approach based detection algorithm. Here it can be seen that the fault becomes apparent around the thirteen hundredth sampling instance when the test score value starts to consistently violate the ninety five percent confidence interval. This is roughly the same time as the local approach based detection algorithm (figure 2.7), but in

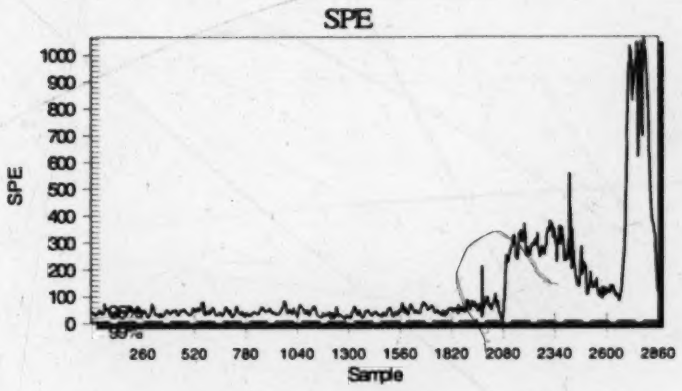

Figure 2.9: SPE plot from PCA on flooding data.

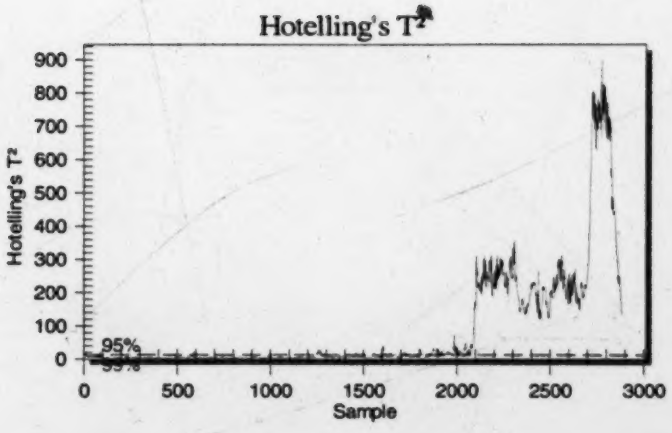

Figure 2.10: Hotelling's T^2 plot from PCA on flooding data.

the SPE plot the test scores do not give as clear picture as the local approach till the two thousandth sampling instance as the test scores are close to the confidence limit bounds. The local approach algorithm on the other hand gives a clearer picture that the fault has occured. Figure 2.10 is the hotelling's T^2 plot got from the PCA analysis. As can be seen it gives a later indication of the flooding with the scores violating the confidence bounds around the eighteen thousandth sampling instance.

2.8 Conclusions

A statistical approach available in the literature for fault detection and isolation called the asymptotic local approach is explained in this chapter. Work done in the literature to increase its applicability is also explained. The application of this approach to various industrial chemical processes with the objective of process monitoring as discussed in this section shows promising results. This approach is found to be sensitive to abrupt process changes which are small in magnitude and can escape detection till it is too late. Also the approach can be used with a variety of modeling techniques for different applications. In the distillation column flooding detection application, the proposed algorithm is able to detect the process fault around three hundred sampling instances before it is visible to the plant personnel. Other industrial examples and simulation examples also show that it is effective for change detection with the potential of detecting faults which are small in magnitude early. Simulation results also show the robustness of the local approach towards issues like underparameterization. This makes it attractive for practical applications where modeling errors and reduced order models are commonplace. The local approach is also compared to monitoring approaches based on SVD and PCA. The results shows that the local approach gave better results than the SVD based approach for the cases considered. The results obtained in the PCA based approach is comparable as shown in the SPE plot, though the SPE plot shows more false alarms. The results obtained from the local approach can also be said to be better if compared with the T^2 plot.

Chapter 3

Control Relevant Model Validation and Its Application

3.1 Introduction

Model Predictive Controllers (MPC) are becoming widely accepted by the process industry. The basic principal of a MPC is to compute the future control moves by minimizing certain objective function over a finite prediction horizon. A simple illustration is shown in figure 3.1 (Camacho and Bordons 1998). The past control moves and outputs of the process are fed into the model (parametric representation of the plant), to generate the future plant response or the predicted outputs over a horizon. These predictions are compared to some reference trajectory to generate the future prediction errors over the prediction horizon. The role of the optimizer is to devise the future control moves which can minimize these prediction errors over the prediction horizon while taking into account the constraints and a certain cost function. The first of these control moves, generated over the prediction horizon, is implemented (receding horizon). Figure 3.2 also shows the same logic but in a graphical representation, the MPC here is sitting at time 'k', reads past values of the output or controlled variable 'y' and the manipulated variable 'u'. The model then predicts the future outputs (over the prediction horizon, time 'k' to '$k+p$' in the figure 3.2), the optimizer then devises moves of the manipulative variable to minimize the difference between 'y' and the 'target' over the prediction horizon.

The recent popularity of these model based predictive controllers can be attributed to the ability to handle constraints and interactions between different variables among other things, which makes them more attractive than simple PID type controllers.

From the brief overview of the MPC's given it is obvious that the model of the process being controlled forms a very important part of the controller. The control moves devised by the optimizer heavily rely on the predictions generated by the model. Hence the identification and validation of this model are very important to ensure the optimality of the MPC. This section deals with these two issues, some of the material in section 3.2 can be seen in Huang and Malhotra (2000) and has been included in detail here to explain how

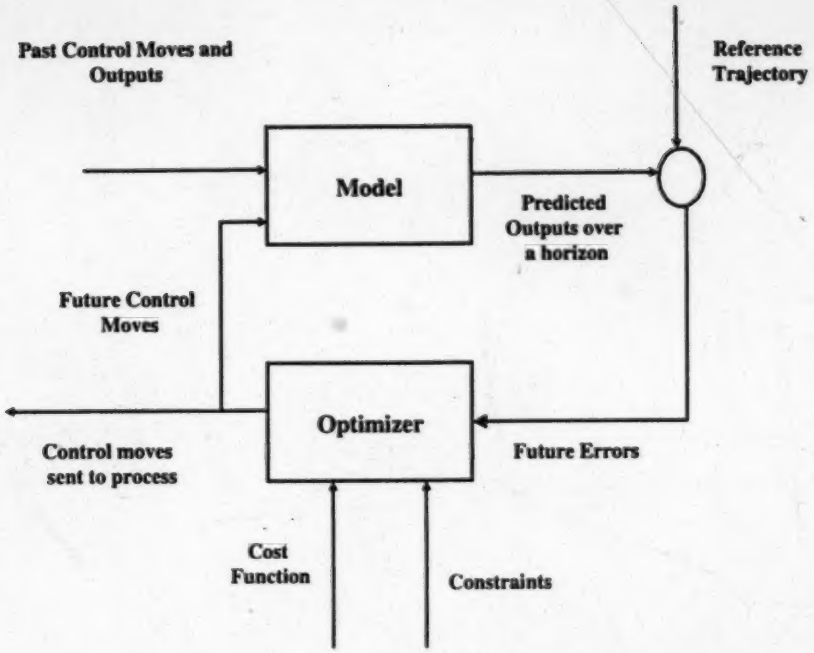

Figure 3.1: Basic structure of a Model Predictive Controller.

data prefiltering can be used for predictive control relevant identification. This concept of data prefiltering will be used in section 3.3 to evolve a strategy for predictive control relevant model validation. A simulation example in given in section 3.4. Section 3.5 deals with application of the algorithm on industrial data. Finally conclusions are discussed in section 3.6.

3.2 Data prefiltering for control relevant identification

For the past twenty years, identification and controller design have developed along two separate directions with hardly any relationship (Van den Hof and Schrama 1995). For controller design, one typically assumes that model and uncertainty are given a priori. On the other hand, studies on identification have focused on questions of convergence and efficiency of the parameter and transfer function estimates in the case when the true system is contained in the model set.

Under such an assumption, the most statistically efficient identification method can be found using the maximum likelihood estimation. The prediction error method has been shown to be asymptotically the same as the maximum likelihood method (Soderstrom and Stoica 1989). It therefore gives statistically the most efficient estimator (Sternad and Soderstrom 1988). The prediction error method is also attractive in closed-loop identifi-

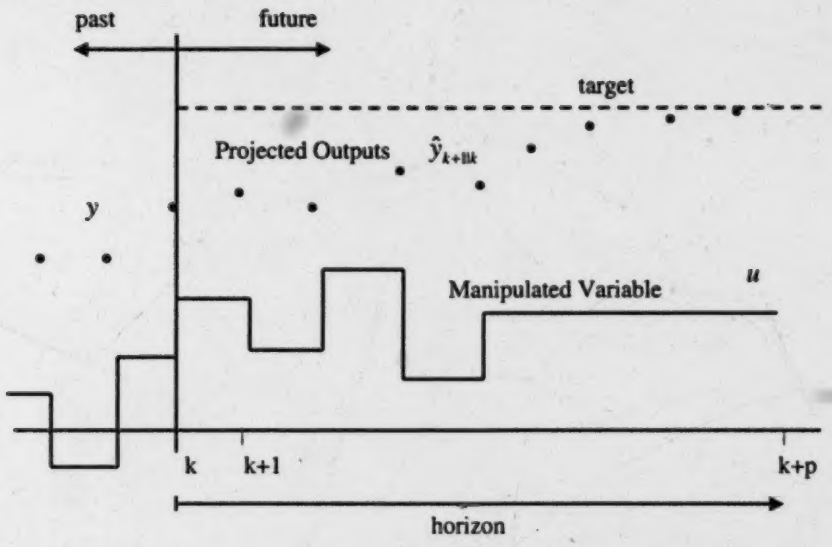

Figure 3.2: Time domain trajectory of the actions of a Model Predictive Controller.

cation where it can give consistent estimates of both process and noise models. Another important feature of the prediction error method is its clear explanation of the model uncertainties in the frequency domain for the design of robust control. Due to its significant advantages over other identification methods, the prediction error method has become the standard identification tool (Ljung 1997, Soderstrom and Stoica 1989, Bitmead *et al.* 1990).

However, the more typical case is that of under-modeling or identification of reduced-complexity models when the plant is not in the model set (Van den Hof and Schrama 1995). Under such circumstances, the model-plant mismatch often consists of both bias error (due to under-modeling) and variance error (due to disturbances). Control performance has to be compromised to accommodate these model uncertainties, and direct application of the prediction error method will not give the best model for control.

A control relevant identification scheme that aims at achieving optimal control performance has been proposed by many researchers (Zang *et al.* 1995, Van den Hof and Schrama 1995). Control relevant identification is to search for appropriate data prefilters before applying the prediction error method such that the identified model is most suitable for the controller design. If we express the model in the frequency domain, then data prefiltering is equivalent to changing the weight of model errors in the identification objective function. Larger weights in some frequencies generally result in less error of the model in the corresponding frequencies. However, this typically increases model error in some other

frequencies and is known as the water-bed effect. Therefore, the data prefilter should be chosen in such a way that yields the most accurate model fitting on the frequencies that are critical to control performance.

The role of data prefiltering has been extensively discussed in the literature. Fundamental theory on data prefiltering can be found in the pioneering work of Ljung (1997). Recently, data prefiltering for the LQG controller design has been discussed by Bitmead *et al.*(1990). Most recently, the role of data prefiltering for closed-loop identification has been discussed by Huang and Shah (1997) , and a new two-step open-loop equivalent closed-loop identification method has been developed by using appropriate data prefilters.

A notable work of identifying the data prefilter for model predictive control can be found in Shook *et al.*(1992) . It is argued that the best model for long range predictive control should be able to provide the best prediction over the prediction horizon. The long range predictive identification (LRPI) as proposed by Shook *et al.*(1992) serves for such a purpose. However, the method developed by Shook *et al.*(1992) is limited to open-loop condition. It has been shown by many researchers (Hjalmarsson *et al.* 1994, Zang *et al.* 1995, Van den Hof and Schrama 1995) that for the design of the LQG controller, the "best" model is obtained under closed-loop condition. The natural question here is whether this conclusion is also true for the model predictive control that is similar to the finite horizon LQ control law. If it is true, what would be the appropriate data prefilter and experiment conditions for such identification ? These questions will be discussed in the following two sections.

3.2.1 Data prefiltering for optimal k-step ahead prediction model identification

In this section the work of Huang *et al* (2000) on optimal k-step ahead model identification using data prefiltering is presented.

Ljung (1997) has shown that for the general Box-Jenkins model

$$y_t = G_p u_t + G_L a_t \tag{3.1}$$

where G_p and G_L are process and disturbance transfer functions respectively and a_t is white noise, the optimal one-step ahead predictor is given by

$$\hat{y}_{t|t-1} = G_L^{-1} G_p u_t + (1 - G_L^{-1}) y_t \tag{3.2}$$

Note that G_p and G_L are not known in general. If the estimated models $\hat{G}_p$ and $\hat{G}_L$ are used, then the estimated optimal predictor may be written as

$$\hat{y}_{t|t-1} = \hat{G}_L^{-1} \hat{G}_p u_t + (1 - \hat{G}_L^{-1}) y_t \tag{3.3}$$

The one-step ahead prediction error is given by

$$\begin{aligned}\epsilon_{t|t-1} &= y_t - \hat{\hat{y}}_{t|t-1} \\ &= \frac{1}{\hat{G}_L}(y_t - \hat{G}_p u_t) \\ &= \frac{1}{\hat{G}_L}[(G_p - \hat{G}_p)u_t + G_L a_t)] \\ &= \frac{1}{\hat{G}_L}[\tilde{G}_p, G_L]\begin{bmatrix} u_t \\ a_t \end{bmatrix}\end{aligned}$$

where $\tilde{G}_p = G_p - \hat{G}_p$. It follows from the Parseval's Theorem that

$$E[\epsilon^2_{t|t-1}] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{|\hat{G}_L(e^{j\omega})|^2}\begin{bmatrix} \tilde{G}_p(e^{j\omega}) & G_L(e^{j\omega}) \end{bmatrix}\begin{bmatrix} \Phi_u(\omega) & \Phi_{ua}(\omega) \\ \Phi_{au}(\omega) & \sigma_a^2 \end{bmatrix}\begin{bmatrix} \tilde{G}_p(e^{-j\omega}) \\ G_L(e^{-j\omega}) \end{bmatrix} d\omega \tag{3.4}$$

The optimal estimates, $\hat{G}_p$ and $\hat{G}_L$, that minimize the prediction error may be obtained by minimizing this prediction error equation, i.e.

$$[\hat{G}_p, \hat{G}_L] = \text{argmin}\frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{|\hat{G}_L(e^{j\omega})|^2}\begin{bmatrix} \tilde{G}_p(e^{j\omega}) & G_L(e^{j\omega}) \end{bmatrix}\begin{bmatrix} \Phi_u(\omega) & \Phi_{ua}(\omega) \\ \Phi_{au}(\omega) & \sigma_a^2 \end{bmatrix}\begin{bmatrix} \tilde{G}_p(e^{-j\omega}) \\ G_L(e^{-j\omega}) \end{bmatrix} d\omega \tag{3.5}$$

This is known as the prediction error method. One may see from the derivation that estimates obtained from the prediction error method should give the best one-step ahead prediction. Equation 3.3 can be extended to the k-step ahead predictor (Ljung 1997):

$$\hat{\hat{y}}_{t|t-k} = \hat{W}_k \hat{G}_p u_t + (1 - \hat{W}_k) y_t \tag{3.6}$$

where

$$\hat{W}_k = \hat{F}_k \hat{G}_L^{-1} \tag{3.7}$$

and

$$\hat{F}_k = \sum_{i=0}^{k-1} g(i) q^{-i} \tag{3.8}$$

where $g(i)$ is the impulse response coefficient of $\hat{G}_L$. The k-step ahead prediction error is given by

$$\begin{aligned}\epsilon_{t|t-k} &= y_t - \hat{\hat{y}}_{t|t-k} \\ &= \hat{W}_k(y_t - \hat{G}_p u_t) \\ &= \frac{\hat{F}_k}{\hat{G}_L}[(G_p - \hat{G}_p)u_t + G_L a_t)] \qquad (3.9) \\ & \qquad (3.10)\end{aligned}$$

Following the same procedure as the derivation of equation (3.5), the estimates, $\hat{G}_p$ and $\hat{G}_L$, that minimize the k-step ahead prediction error, can be derived as

$$[\hat{G}_p, \hat{G}_L] = \text{argmin} \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{|\hat{F}_k(e^{j\omega})|^2}{|\hat{G}_L(e^{j\omega})|^2} \left[\begin{array}{cc} \bar{G}_p(e^{j\omega}) & G_L(e^{j\omega}) \end{array} \right] \left[\begin{array}{cc} \Phi_u(\omega) & \Phi_{ua}(\omega) \\ \Phi_{au}(\omega) & \sigma_a^2 \end{array} \right] \left[\begin{array}{c} \bar{G}_p(e^{-j\omega}) \\ G_L(e^{-j\omega}) \end{array} \right] d\omega \tag{3.11}$$

Similarly, the estimates calculated by minimizing the k-step ahead prediction error should give the best k-step prediction. If we filter y_t and u_t by $\hat{F}_k$ and denote the filtered data as y_t^f and u_t^f, then the one-step ahead predictor of y_t^f (by fixing the noise model as $\hat{G}_L$) is given by

$$\hat{\hat{y}}_{t|t-1}^f = \hat{G}_L^{-1}\hat{G}_p u_t^f + (1 - \hat{G}_L^{-1}) y_t^f$$

The one-step ahead prediction error of y_t^f is therefore given by

$$\begin{aligned} \epsilon_{t|t-1}^f &= y_t^f - \hat{\hat{y}}_t^f \\ &= \frac{1}{\hat{G}_L}(y_t^f - \hat{G}_p u_t^f) \\ &= \frac{\hat{F}_k}{\hat{G}_L}(y_t - \hat{G}_p u_t) \\ &= \frac{\hat{F}_k}{\hat{G}_L}[(G_p - \hat{G}_p)u_t + G_L a_t)] \end{aligned}$$

which is the same as equation (3.9), i.e. k-step ahead prediction error is the same as one-step ahead prediction error of the filtered data. Therefore, the estimates calculated from equation (3.11) can be obtained using the prediction error method (one-step ahead prediction) of the filtered data by fixing the noise model. We shall call this method as the k-step prediction error method or KPEM (Huang and Malhotra 2000).

3.2.2 Data prefiltering for optimal multistep prediction model identification

In this section the work of Huang *et al* (2000) on optimal multi-step ahead model identification using data prefiltering is presented.

If we are interested in optimal multistep prediction over a finite prediction horizon from N_1 to N_2, the objective function may be defined as

$$J_{multi} = \frac{1}{N_p} \sum_{i=N_1}^{N_2} E[y_{t+i} - \hat{\hat{y}}_{t+i|t}]^2 \tag{3.12}$$

where $N_p = N_2 - N_1 + 1$ is the prediction horizon and $E[y_{t+i} - \hat{\hat{y}}_{t+i|t}]^2$ is the i step ahead prediction error. Rewrite equations (3.6) and (3.7) as

$$\hat{\hat{y}}_{t|t-i} = \hat{F}_i \hat{G}_L^{-1}\hat{G}_p u_t + (1 - \hat{F}_i \hat{G}_L^{-1}) y_t \tag{3.13}$$

or

$$\hat{y}_{t+i|t} = \hat{F}_i\hat{G}_L^{-1}\hat{G}_p u_{t+i} + (1 - \hat{F}_i\hat{G}_L^{-1})y_{t+i} \tag{3.14}$$

Note that y_{t+i} and u_{t+i} satisfy the model

$$y_{t+i} = G_p u_{t+i} + G_L a_{t+i} \tag{3.15}$$

From equations (3.14) and (3.15), it follows that

$$\begin{aligned} y_{t+i} - \hat{\hat{y}}_{t+i|t} &= \hat{F}_i\hat{G}_L^{-1}y_{t+i} - \hat{F}_i\hat{G}_L^{-1}\hat{G}_p u_{t+i} \\ &= \hat{F}_i\hat{G}_L^{-1}(G_p u_{t+i} + G_L a_{t+i}) - \hat{F}_i\hat{G}_L^{-1}\hat{G}_p u_{t+i} \\ &= \hat{F}_i\hat{G}_L^{-1}\begin{bmatrix} \tilde{G}_p & G_L \end{bmatrix}\begin{bmatrix} u_{t+i} \\ a_{t+i} \end{bmatrix} \end{aligned}$$

Applying the Parseval's Theorem yields

$$E[y_{t+i} - \hat{\hat{y}}_{t+i|t}]^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{|\hat{F}_i(e^{j\omega})|^2}{|\hat{G}_L(e^{j\omega})|^2}\begin{bmatrix} \tilde{G}_p(e^{j\omega}) & G_L(e^{j\omega}) \end{bmatrix}\begin{bmatrix} \Phi_u(\omega) & \Phi_{ua}(\omega) \\ \Phi_{au}(\omega) & \sigma_a^2 \end{bmatrix}\begin{bmatrix} \tilde{G}_p(e^{-j\omega}) \\ G_L(e^{-j\omega}) \end{bmatrix} d\omega \tag{3.16}$$

Substituting equation (3.16) into equation (3.12) yields

$$J_{multi} = \frac{1}{N_p}\sum_{i=N_1}^{N_2}\frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{|\hat{F}_i(e^{j\omega})|^2}{|\hat{G}_L(e^{j\omega})|^2}\begin{bmatrix} \tilde{G}_p(e^{j\omega}) & G_L(e^{j\omega}) \end{bmatrix}\begin{bmatrix} \Phi_u(\omega) & \Phi_{ua}(\omega) \\ \Phi_{au}(\omega) & \sigma_a^2 \end{bmatrix}\begin{bmatrix} \tilde{G}_p(e^{-j\omega}) \\ G_L(e^{-j\omega}) \end{bmatrix} d\omega \tag{3.17}$$

The summation may be moved within the integral. This gives

$$J_{multi} = \frac{1}{N_p}\frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sum_{i=N_1}^{N_2}|\hat{F}_i(e^{j\omega})|^2}{|\hat{G}_L(e^{j\omega})|^2}\begin{bmatrix} \tilde{G}_p(e^{j\omega}) & G_L(e^{j\omega}) \end{bmatrix}\begin{bmatrix} \Phi_u(\omega) & \Phi_{ua}(\omega) \\ \Phi_{au}(\omega) & \sigma_a^2 \end{bmatrix}\begin{bmatrix} \tilde{G}_p(e^{-j\omega}) \\ G_L(e^{-j\omega}) \end{bmatrix} d\omega \tag{3.18}$$

Minimization of equation (3.18) gives the estimates, $\hat{G}_p$ and $\hat{G}_L$,

$$\begin{aligned} [\hat{G}_p, \hat{G}_L] = \text{argmin}\frac{1}{N_p}\frac{1}{2\pi}\int_{-\pi}^{\pi} &\frac{\sum_{i=N_1}^{N_2}|\hat{F}_i(e^{j\omega})|^2}{|\hat{G}_L(e^{j\omega})|^2}\begin{bmatrix} \tilde{G}_p(e^{j\omega}) & G_L(e^{j\omega}) \end{bmatrix} \\ &\times \begin{bmatrix} \Phi_u(\omega) & \Phi_{au}(\omega) \\ \Phi_{ua}(\omega) & \sigma_a^2 \end{bmatrix}\begin{bmatrix} \tilde{G}_p(e^{-j\omega}) \\ G_L(e^{-j\omega}) \end{bmatrix} d\omega \end{aligned} \tag{3.19}$$

that minimize the multistep prediction error. By comparing equation (3.18) to the one step prediction error in equation (3.4) (noticing that the constant $1/N_p$ does not affect the optimization), the difference is the filter

$$\hat{L}_{N_1,N_2} = \sum_{i=N_1}^{N_2}|\hat{F}_i(e^{j\omega})|^2 \tag{3.20}$$

which, when cascaded to the one step ahead prediction, yields the multistep prediction. Therefore, the estimates that minimize multistep prediction error may be calculated from the filtered one-step ahead prediction error method by fixing the noise model. We shall call this method as multistep prediction error method (MPEM). The following algorithm is suggested for the multistep prediction error method.

Algorithm 1 *1. Use the prediction error method to get initial estimates, $\hat{G}_p$ and $\hat{G}_L$.*

2. Filter both y_t and u_t by the multistep prediction filter $\hat{L}_{N_1,N_2}$, where $\hat{L}_{N_1,N_2}$ can be calculated from $\hat{G}_L$ according to equations 3.20 and 3.8.

3. With the fixed noise model $\hat{G}_L$, apply the prediction error method to the filtered data y_t^f and u_t^f. This gives the estimate $\hat{G}_p$ that minimizes the multistep prediction error.

Clearly, the multistep prediction error method derived in this section is model predictive control relevant. Model predictive control typically requires a model to provide multistep predictions over a finite horizon. For example, the objective of identification for MPC is to provide a model that can give good predictions over a finite prediction horizon from N_1 to N_2. In addition, as one can see from the above derivations that we have never assumed $\Phi_{ua}(\omega) = 0$, the KPEM and MPEM identification methods can be used under both closed-loop and open-loop conditions.

3.3 KPEM and MPEM relevant model validation

Now let us parameterize a process model by

$$y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}a(t) \tag{3.21}$$

where

$$\frac{B(q^{-1})}{F(q^{-1})} = G_P$$

$$\frac{C(q^{-1})}{D(q^{-1})} = G_L$$

are process and disturbance transfer functions respectively, and

$$\begin{aligned} B(q^{-1}) &= b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b} \\ F(q^{-1}) &= f_1 q^{-1} + \cdots + f_{n_f} q^{-n_f} \\ C(q^{-1}) &= c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c} \\ D(q^{-1}) &= d_1 q^{-1} + \cdots + d_{n_d} q^{-n_d} \end{aligned}$$

The parameter of interest, θ, is given by

$$\theta = [\ b_1 \ \cdots \ b_{n_b} \ \ f_1 \ \cdots \ f_{n_f} \ \ c_1 \ \cdots \ c_{n_c} \ \ d_1 \ \cdots \ d_{n_d}\ ]^T$$

Model validation is equivalent to the detection of parameter changes (Benveniste *et al.* 1987). The objective of this section is to derive a local approach based detection algorithm that is relevant to the KPEM or MPEM objective. That is, the detection algorithm should only be sensitive to the parameter changes that can affect the k-step ahead prediction or multistep ahead prediction. The solution may be found by solving the PEM detection problem since the KPEM and MPEM are equivalent to the filtered PEM . Let us first derive the local approach detection algorithm that is PEM objective relevant. Then we will extend the result to the KPEM or MPEM relevant detection algorithm in a straightforward manner. The prediction error method (PEM) is to minimize the following objective function (Ljung 1997)

$$J = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}e_{t|t-1}^2 \tag{3.22}$$

where $e_{t|t-1}$ is the one step ahead prediction error given by

$$e_{t|t-1} = y_t - \hat{y}_{t|t-1}$$

and $\hat{y}_{t|t-1}$ is the one step ahead prediction given by

$$\hat{y}_{t|t-1} = (\frac{D(q^{-1})}{C(q^{-1})})\frac{B(q^{-1})}{F(q^{-1})}u_t + (1 - \frac{D(q^{-1})}{C(q^{-1})})y_t$$

The gradient of equation 3.22 can be calculated as

$$\frac{\partial J}{\partial \theta} = -\frac{1}{N}\sum_{t=1}^{N}(\frac{\partial e_{t|t-1}}{\partial \theta})^T e_{t|t-1} \tag{3.23}$$

Assume that θ_0 is calculated by equating equation 3.23 to zero. Then the estimation θ_0 based on the prediction error method that minimizes equation 3.22 must satisfy

$$\frac{\partial J}{\partial \theta}|_{\theta_0} = -[\frac{1}{N}\sum_{t=1}^{N}(\frac{\partial e_{t|t-1}}{\partial \theta})^T e_{t|t-1}]_{\theta=\theta_0} = 0 \tag{3.24}$$

As the sample size increases, equation 3.24 is asymptotically equivalent to

$$E[(\frac{\partial e_{t|t-1}}{\partial \theta})^T e_{t|t-1}]_{\theta=\theta_0} = 0 \tag{3.25}$$

Obviously, for any new data generated from equation 3.21, if $\theta = \theta_0$, then equation 3.25 should continue to hold. If, on the other hand, $\theta \neq \theta_0$ then equation 3.25 will be non-zero. Hence, the model validation problem is reduced to that of monitoring the mean of a vector which is given by:

$$H(\theta_0, \phi_t) = (\frac{\partial e_{t|t-1}}{\partial \theta})^T e_{t|t-1} \tag{3.26}$$

This is exactly the primary residual as defined in equations (2.9) and (2.10) in chapter 2. The actual monitoring can be done by performing a χ^2 test on the normalized residual given by:

$$\xi_N(\theta_0) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} H(\theta_0, \phi_t) \tag{3.27}$$

To calculate the primary residual, the partial derivative of the prediction error needs to be calculated first. The partial derivative of the prediction errors can be calculated as:

$$(\frac{\partial e_{t|t-1}}{\partial \theta}) = (\frac{\partial \hat{y}_{t|t-1}}{\partial \theta}) = \psi(t, \theta) \tag{3.28}$$

where:

$$\psi(t, \theta) = \frac{\partial \hat{y}_{t|t-1}}{\partial \theta} = [\frac{\partial}{\partial b_k}\hat{y}_{t|t-1} \quad \frac{\partial}{\partial c_k}\hat{y}_{t|t-1} \quad \frac{\partial}{\partial d_k}\hat{y}_{t|t-1} \quad \frac{\partial}{\partial f_k}\hat{y}_{t|t-1}]^T \tag{3.29}$$

The partial derivatives of $\hat{y}_{t|t-1}$ with respect to the parameters can be evaluated from equation 3.21 as (Ljung 1997):

$$\begin{aligned}
\frac{\partial}{\partial b_k}\hat{y}_{t|t-1} &= -\frac{D(q^{-1})}{C(q^{-1})F(q^{-1})}u_{t-k} \\
\frac{\partial}{\partial c_k}\hat{y}_{t|t-1} &= -\frac{D(q^{-1})B(q^{-1})}{C(q^{-1})C(q^{-1})F(q^{-1})}u_{t-k} + \frac{D(q^{-1})}{C(q^{-1})C(q^{-1})}y_{t-k} \\
\frac{\partial}{\partial d_k}\hat{y}_{t|t-1} &= -\frac{B(q^{-1})}{C(q^{-1})F(q^{-1})}u_{t-k} - \frac{1}{C(q^{-1})}y_{t-k} \\
\frac{\partial}{\partial f_k}\hat{y}_{t|t-1} &= -\frac{D(q^{-1})B(q^{-1})}{C(q^{-1})F(q^{-1})F(q^{-1})}u_{t-k}
\end{aligned}$$

For the simple case of an output error (OE) model

$$y_t = \frac{B(q^{-1})}{F(q^{-1})} + v_t$$

where the disturbance model is not parameterized, the expression for $\psi(t, \theta)$ reduces to:

$$\psi(t, \theta) = \frac{\phi(t, \theta)}{F(q^{-1})} \tag{3.30}$$

where:

$$\phi(t, \theta) = \begin{bmatrix} u_{t-1} & \cdots & u_{t-n_b} & -\hat{y}_{t-1|t} & \cdots & -\hat{y}_{t-n_f|t-n_f+1} \end{bmatrix}^T \tag{3.31}$$

Therefore, the primary residual for the PEM case is

$$H(\theta_0, \phi_t) = \psi(t, \theta_0)e_{t|t-1} \tag{3.32}$$

and for the OE case where $C(q^{-1})$ and $D(q^{-1})$ are not parameterized becomes

$$H(\theta_0, \phi_t) = \frac{\phi(t, \theta_0)}{F(q^{-1})}e_{t|t-1} \tag{3.33}$$

In many applications, it may be desired that the model validation algorithm should not give an alarm if parameter changes occur only in the disturbance models. It has been demonstrated that by associating the detection algorithm with the output error method, the model validation algorithm based on the local approach is disturbance model independent (Huang 2000). We shall call this method as the output error method (OEM) based local approach. To summarize, model validation can be performed using the following algorithm:

Algorithm 2 *Provided that model parameters in θ_0 are identified from a set of identification data, a set of validation data can then be used to perform the following model validation test:*

1. *Calculate the primary residual $H(\theta_0, \phi_t)$ using equation 3.32 or equation 3.33.*
2. *Calculate the normalized residual $\xi_N(\theta_0)$ using equation 3.27.*
3. *Calculate the covariance matrix $\Sigma(\theta_0)$ using equation 2.16.*
4. *Calculate χ^2_{global} using equation 2.18.*
5. *Select the threshold value from the χ^2 table according to a pre specified false alarm rate with the degree of freedom being equal to the row dimension of θ.*
6. *If χ^2_{global} is larger than the threshold, issue an alarm and perform the isolation test; otherwise conclude that the model passes the validation test.*

Remark 3 *There are two scenarios in this test, namely model validation and model change detection. If θ_0 is treated as a set of parameters to be validated, then this test falls in the category of model validation as in the identification literature. In this case, failure of the test means that θ_0 is falsified by the validation data. On the other hand, if θ_0 is treated as a set of initial true parameters, then this test falls in the category of detection of model changes. Failure of the test means that the model parameters have changed from the initial values.*

This method will validate model parameters that affect the one-step ahead prediction (for the PEM case). The result can be extended to the k-step and multi-step prediction by using KPEM or MPEM data prefilters. In the case of the KPEM filter data will be prefiltered by the filter given by $\hat{F}_K$ in equation 3.8. Then the validation algorithm will monitor only those changes or mismatches that have an effect on the k-step ahead prediction. Similarly the extension to the multi-step ahead prediction case is obtained by prefiltering the data using the multistep ahead prediction filter (MPEM) whose spectrum is given by equation 3.20. Then, the model validation algorithm will test only those parameter changes or mismatches which have an effect on the multistep ahead predictions. Needless to say, these algorithms are highly relevant for validation of models that are primarily used for multistep ahead prediction, such as for model predictive control.

3.4 Simulation

To simplify the notation, in this section, we shall use the following convention:

- PEM — (one-step ahead) prediction error method
- OEM — output error method
- KPEM — k-step ahead prediction error method
- MPEM — multistep ahead prediction error method
- KOEM — k-step ahead output error method
- MOEM — multistep ahead output error method

Note that the primary difference between the prediction error method based detection algorithms (PEM, KPEM, MPEM) and output error method based detection algorithms (OEM, KOEM, MOEM) is whether the detection algorithm is sensitive to the changes of the disturbance transfer functions. In addition, the prediction error method based identification and detection algorithms can handle random walk disturbance which is unbounded while output error based identification and detection algorithms can not.

The disturbance transfer function is fixed in all the validation algorithms discussed in this section to the random walk:

$$G_L = \frac{1}{1 - q^{-1}} \tag{3.34}$$

This is done as this form of disturbance model is commonly used in model predictive control. It is interesting to see how this model fits practical data sampled from model predictive control systems as this random walk model actually represents an unbounded disturbance while the actual disturbances are typically bounded. The KPEM and MPEM prefilters are formed using this disturbance transfer function (G_L). The normalized magnitude Bode plot of the KPEM filter is given in figure 3.3. The plot shows the weighting distribution over the frequency range of interest. It can be seen that the frequency distribution shifts towards the lower frequency band as the prediction horizon is increased. With increased prediction horizon the frequency plot of the filter approaches that of a pure integrator. Due to this reason the detection algorithm with KPEM filtering would be less sensitive to high frequency model changes with increased prediction horizon. The normalized magnitude Bode plot of the MPEM filter is given in figure 3.4. Here again it can be seen that the frequencies over which the filter acts shifts towards the lower frequency band as the prediction horizon is increased. Also the relative weighting on the higher frequencies decreases as prediction horizon increases.

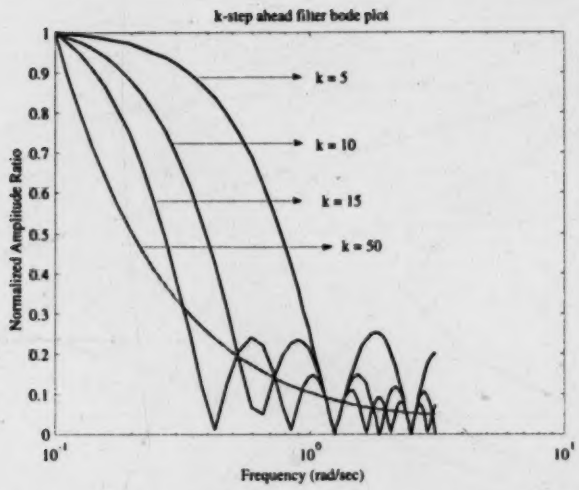

Figure 3.3: normalized amplitude Bode plot of KPEM filter (using 5,10,15 and 50 step).

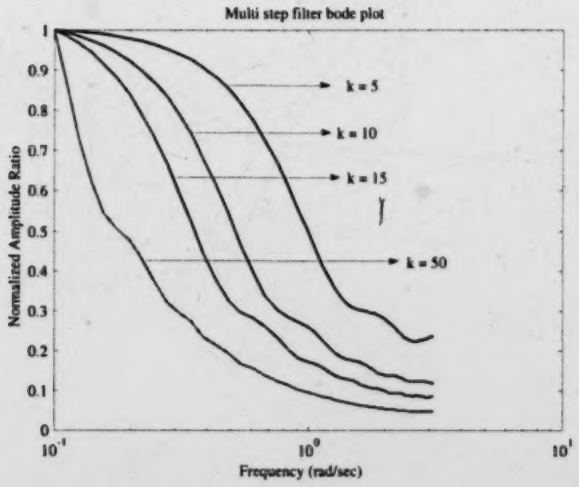

Figure 3.4: normalized amplitude Bode plot of MPEM filter (using 5,10,15 and 50 step).

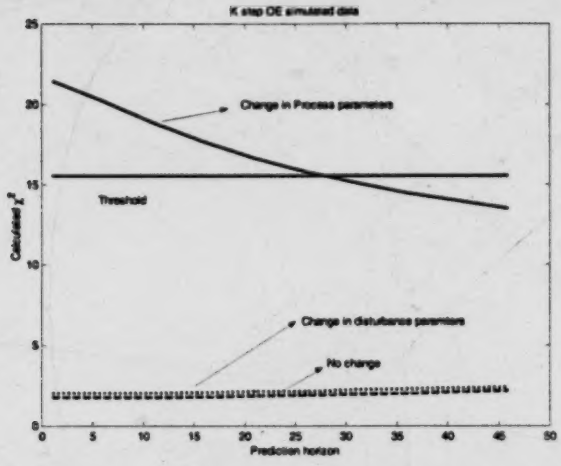

Figure 3.5: plot of χ^2 test using the KOEM based detection algorithm.

Example 3.1

In this example, we apply the proposed control relevant detection algorithms to the process given by equation 3.35. To generate the input and output data we simulate a disturbance model which is very close to the random walk but stable, i.e. the denominator of the disturbance transfer function has a pole 0.99 instead of 1.

$$y_t = \frac{0.00768z^2 + 0.02123z + 0.00357}{z^3 - 1.9031z^2 + 1.1514z - 0.2158}u_{t-1} + \frac{0.01z}{z - 0.99}a_t \tag{3.35}$$

This is a more realistic disturbance which is drifting but bounded. The input (u_t) and noise term (a_t) are taken as zero mean random inputs with variance 0.3 and 0.1 respectively. Different parameter changes are considered to test the effectiveness of the validation algorithms. The results are shown in the following figures where the calculated χ^2 values plot against the prediction horizon. A ten percent change in parameters is taken for each case. Each figure shows the result for three scenarios - no change, change in process parameters and change in disturbance parameters. The threshold from the χ^2 table is also plotted in each figure. Figure 3.5 shows the implementation of the KOEM based validation algorithm. Here it can be seen that a ten percent change in process parameters shows up as a violation of the threshold for the small prediction horizon. Change in the disturbance model is not registered as a threshold violation (which is expected since this algorithm is based on the KOEM formulation and the disturbance model is not being monitored). For the higher prediction horizon the model passes the test indicating that the change in process parameters does not affect the long range predictions. Similar results are obtained in figure 3.6. Here also the changes in the disturbance model are not affecting the validation.

Figures 3.6 shows the implementation of the KPEM based validation algorithm. Here

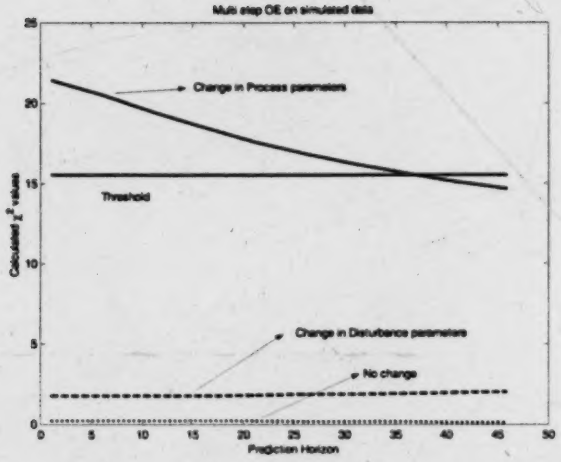

Figure 3.6: plot of χ^2 test using the MOEM based detection algorithm.

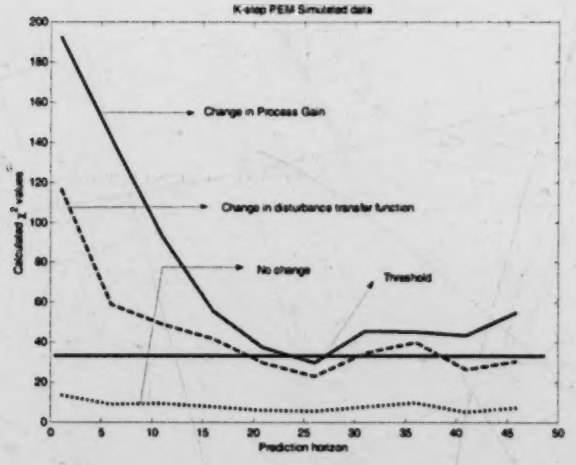

Figure 3.7: plot of χ^2 test using the KPEM based detection algorithm.

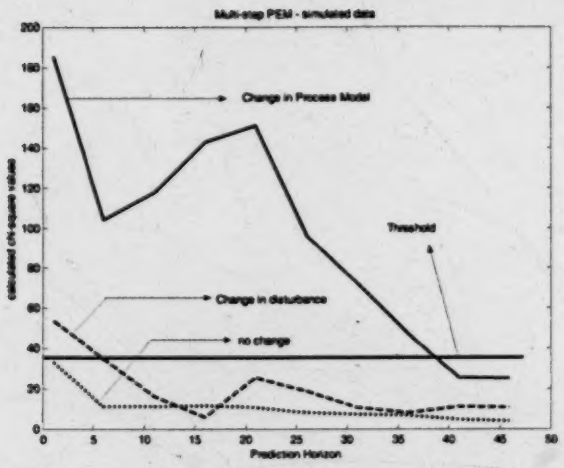

Figure 3.8: χ^2 test using the MPEM based detection algorithm.

again it can be seen that a ten percent change in process parameters shows up as a violation of the threshold for the small prediction horizon. Change in disturbance parameters also show up as violation of the threshold (this again is expected since this algorithm is based on the KPEM formulation and the disturbance model is also monitored). For the higher prediction horizon the model passes the test indicating that the change in process parameters does not affect the long range predictions. Similar results are obtained in figure 3.8 for the case of MPEM filtering.

3.5 Industrial applications

Two industrial case studies are presented in this section. In the first industrial application the algorithm is tried on data sampled from an industrial fluidized coker unit (figure 2.5). The system in question has four outputs or controlled variables (CV's) and two inputs or manipulated variables (MV's). The system is modeled as four MISO systems (each having two inputs). Second order models are identified for the four MISO systems using training data (taken from step test data). The proposed control relevant model validation algorithm is run on a different set of data sampled when the process is in normal operating condition and in the open-loop mode. Whenever all of the four controlled variables are inside their constraints, the MPC completely drops the feedback control to perform economic optimization. The controller is in the open-loop mode and this provides a good opportunity for model validation. Note that the proposed model validation algorithms also work under closed-loop condition if external excitation exists.

The model testing results are presented in the figures below. Each figure in this section shows the calculated χ^2 values plotted against the prediction horizon. Any violation of

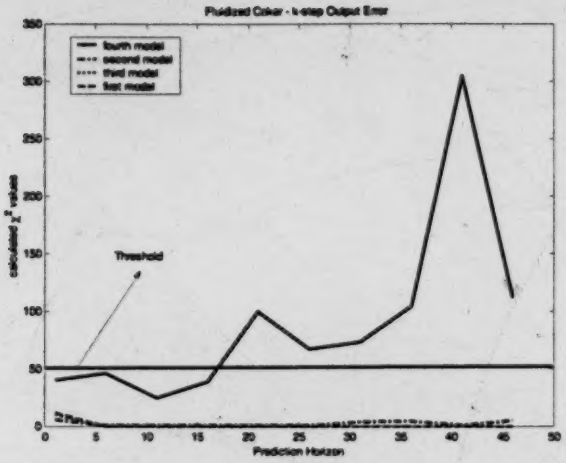

Figure 3.9: χ^2 test using the KOEM filter - Fluidized Coker case study.

the threshold means that the model fails the validation test for that particular prediction horizon. Here again the validation is done using four different filters, namely KPEM, MPEM, KOEM and MOEM. Figures 3.9 shows the result of the model validation based on the KOEM filtering. It can be seen that models corresponding to three outputs pass the test as the calculated χ^2 values are below the threshold. Models corresponding to one of the outputs fails the test and with the increase of prediction horizon the performance of this model can be expected to further degrade, as its violation increases with the increase in the prediction horizon. As shown in figure 3.10, the result using the MOEM based algorithm on the same models shows a slightly different result with one model consistently fails the test, one fails the test for small prediction horizons and one fails the test for large prediction horizons. Figures 3.11 shows the result of the model validation based on KPEM filtering. It can be seen that here only one model passes the test consistently, one fails for small prediction horizons, and two fail consistently. The result for MPEM based algorithm (figure 3.12) on the same models shows a similar result but none of the four models pass the test consistently. The difference between OEM (KOEM, MOEM) and PEM (KPEM, MPEM) based methods is due to the disturbance model: the former is not sensitive to the change in the disturbance model and the latter is sensitive to changes in both process and disturbance models.

The algorithm is also implemented on another industrial application, a hydrogen plant. This hydrogen plant is a large-scale process controlled by an advanced model predictive controller. It has twenty controlled variables, five manipulated variables and nine disturbance variables. For our analysis this system is represented as twenty multiple input single output models (MISO models). To simplify the presentation, validation is performed using

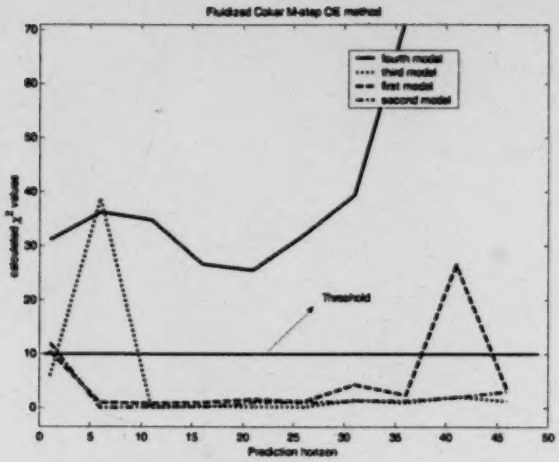

Figure 3.10: χ^2 test using the MOEM filter - Fluidized Coker case study.

the OEM based formulation only:

$$y_t = \sum_{i=1}^{n} \frac{B_i}{F_i} u_t^{(i)} + v_t$$

or

$$y_t = \sum_{i=1}^{n} G_p^{(i)} u_t^{(i)} + v_t$$

where n is the total number of inputs to each output. Validation is performed with KOEM and MOEM based detection algorithms respectively. The calculated χ^2 values are evaluated over prediction horizon ranging from one to fifty.

The results for the two representative controlled variables (these are the first and the sixth controlled variables for the plant being considered denoted by cv_1 and cv_6) are shown in figures 3.13 and 3.14. Here it is seen that cv_1 (figure 3.13) passes the test and cv_6 (figure 3.14) does not pass the test for smaller prediction horizons.

Since figure 3.14 shows that the model fails the tests for smaller prediction horizons the isolation test can be applied to pin point which model is faulty (see the isolation algorithm in section 2.5). The isolation is carried at prediction horizon of fifteen. The plot given by the isolation is shown in figure 3.15. The isolation figure is shown as a bar plots with the model number on the x-axis showing the contributing models ($cv6$ has two contributing manipulated variables only) with a maximum bar indicating a possible faulty model.

The isolation test plot shows that the first contributing model or the model corresponding to the first input.

A similar analysis on the models corresponding to another controlled variable of the same plant (denoted by cv_5) can be seen in figure 3.16. Here it can be seen that the model

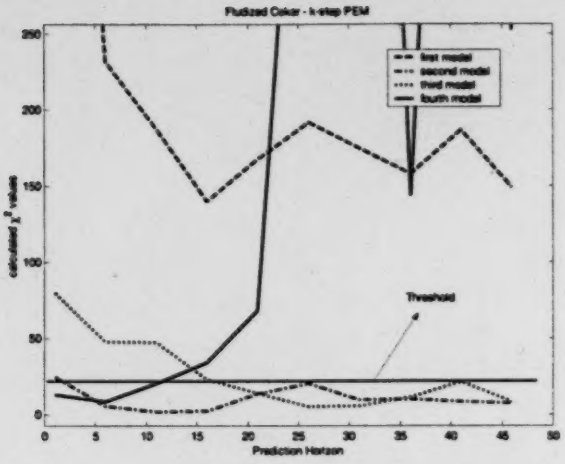

Figure 3.11: χ^2 test using the KPEM filter - Fluidized Coker case study.

fails the test for all prediction horizons. The isolation is also carried out for this case with prediction horizon of ten. The plot given by the isolation is shown in figure 3.17. The isolation figure is shown as a bar plots with the model number on the x-axis showing the contributing models. Here it can be seen that the model corresponding to the tenth input is the major faulty model.

3.6 Conclusion

The role of data prefiltering for model identification and validation has been discussed in this chapter. The k-step and multistep data prefilters are derived from the prediction error method using the general process model. The result is applicable to both open-loop and closed-loop identification with external excitation. The proposed predictive control relevant model validation algorithm detects only those parameters changes which affect the k-step or multi-step ahead prediction. If the controller (minimum variance controller or model predictive controller) uses these predictions to derive its control moves then the validation algorithm derived here is highly relevant. The results from the simulation example and two industrial case studies show that it may be imprudent to reject a model by just validating it with respect to one step ahead predictions or infinite step ahead prediction. In fact models should be validated on the basis of their application objectives. In this chapter, models are validated on the basis of different prediction horizons and it is observed that some models which did not pass the validation test for shorter prediction horizons, passed the test for longer prediction horizons. In some cases the opposite situations are also observed. This is due to the data prefiltering which places larger weightings on the frequencies of interest according to a specific objective or prediction horizon. The changes in the model which is

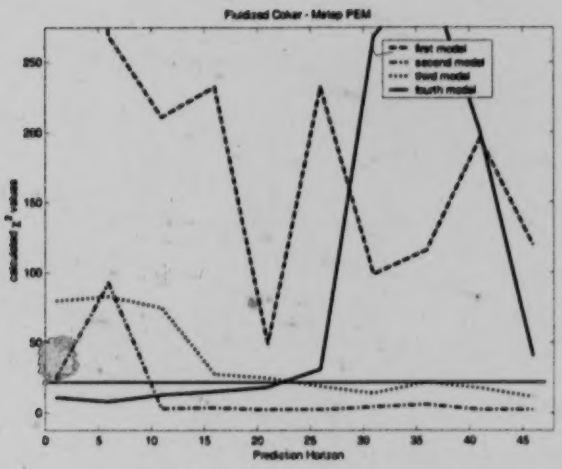

Figure 3.12: χ^2 test using the MPEM filter - Fluidized Coker case study.

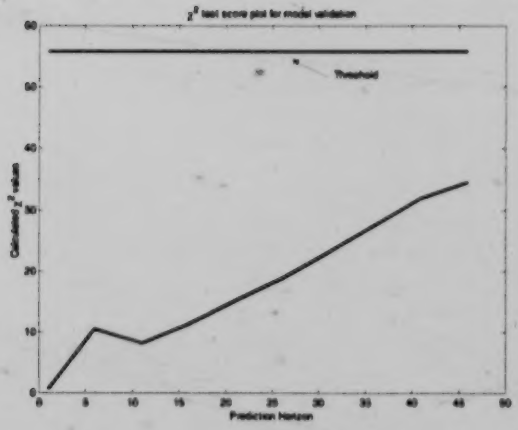

Figure 3.13: χ^2 plotted against the prediction horizons - Hydrogen plant case study - cv_1.

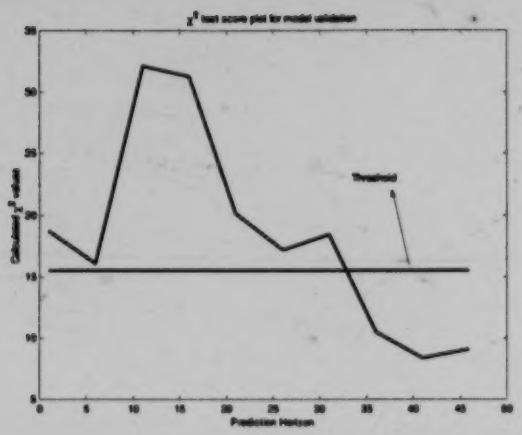

Figure 3.14: χ^2 plotted against the prediction horizons - Hydrogen plant case study - cv_6.

highly weighted by the data prefilter will be easily detected. The change in parameters in other frequencies (other than those highly weighted by the filter) will be in fact suppressed.

The algorithm derived for control relevant model validation in this chapter has been programmed into Matlab/GUI and has been implemented in Syncrude, Upgrading at Ft. McMurray (refer Appendix A for the user manual of the GUI).

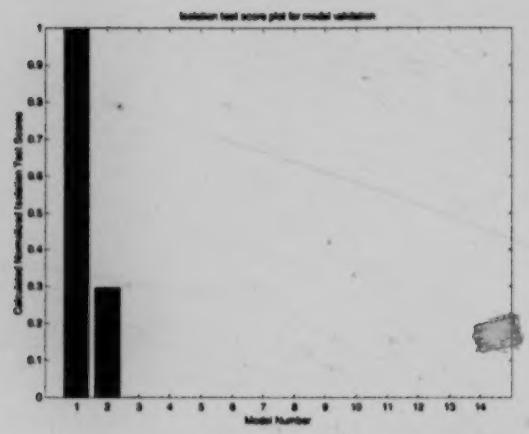

Figure 3.15: Result of isolation tests carried on cv_6.

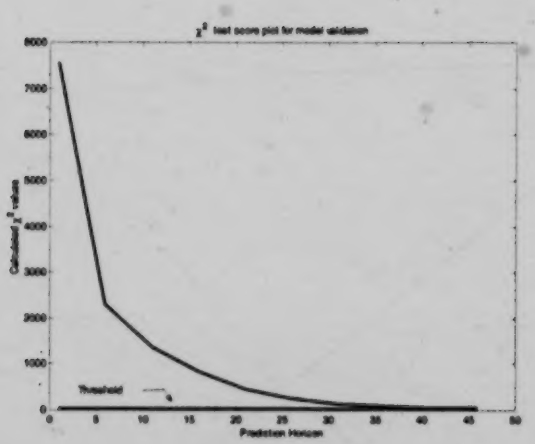

Figure 3.16: χ^2 test scores plotted against the prediction horizons for cv_5.

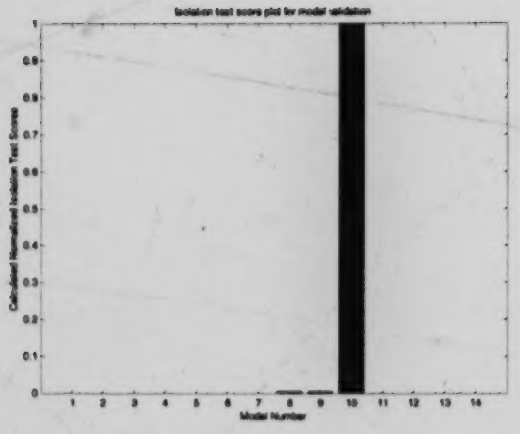

Figure 3.17: Result of Isolation tests carried on cv_5 for ten prediction horizons.

Chapter 4

Monitoring of Process Gain Decalibration Without the Knowledge of Inputs

4.1 Introduction

A typical industrial plant includes hundreds or even thousands of control loops. Every single control loop consists of at least four basic components: the controller, actuator, sensor and the process itself. Instrumentation technicians generally maintain and service these loops. Routine maintenance of such loops at optimal settings can save a typical chemical process industry hundreds of thousands of dollars a year. This routine maintenance can be supplemented by condition based or preventive maintenance through implementation of process monitoring algorithm. This means that in the case where the monitoring algorithm detects a fault in its early stages, preventive steps can be taken before the fault increases in magnitude and affects the process. The control literature has been relatively rich on studies concerned with the subject of process monitoring for early fault detection. However most of the existing methods rely heavily on some input output relation or the model of the process under consideration. The monitoring scheme consists of checking if this representation or model holds for future data. In a regular system since the input is typically known, the dynamic response of the system can relatively easily be assessed by comparing the relation between the output and the input. This relationship is typically modeled by a dynamic time series model or a state observer. This model or observer mimics the dynamic behavior of the process. By monitoring the innovation sequence or residuals generated from the model or observer, one can detect if there is a fault in the process. Thus, this model or observer is the key to process monitoring and fault detection. Since this model or observer is built upon the relation between the input and the output, both of which must be known in order to apply any of these monitoring techniques.

As compared to systems where the inputs are known there is little work done on monitoring of systems where there is no knowledge of the inputs. A typical example is the

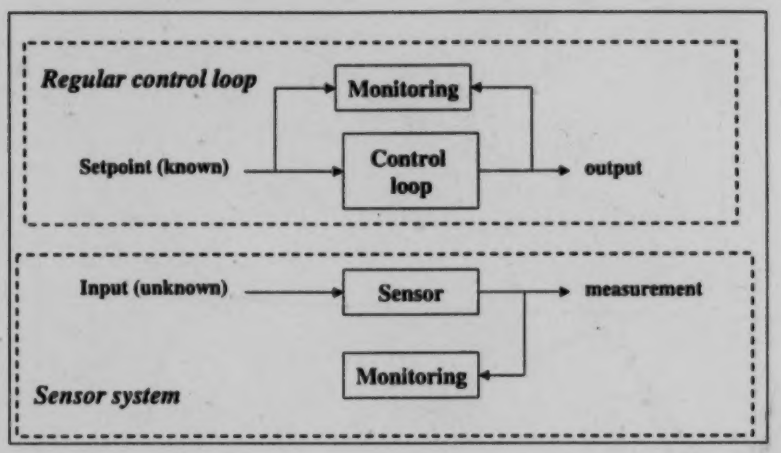

Figure 4.1: The inherent difficulty in sensor monitoring as compared to process or control loop monitoring

monitoring of sensors for decalibration and faults. Sensors are critical elements in process control and automation. A sensor giving wrong information to a controller can have a snow balling effect on the rest of the process as undesired and unnecessary control moves may be implemented based on these incorrect signals. Sensor faults can cause deterioration of productivity due to poor controller performance even though the controller may be optimally tuned. This can trigger many control problems. There have been many instances in the chemical process industries where large scale processes have been forced to have an unscheduled shutdown due to one critical sensor giving faulty readings which devastated the normal operation of the entire process.

To understand this issue and the inherent problem in sensor monitoring as compared to process or controller monitoring let us consider the illustration in figure 4.1. Here a sensor is treated as a system. The output of this system is the measurement of its input, which is an unknown quantity unless the input and the output are exactly same. For a healthy sensor, the output follows the input closely subject to measurement noise/disturbances. If there is a change in the unknown input, then the output should track the change quickly. This 'unknown' nature of the input signal makes sensor monitoring fundamentally different from regular process or control loop monitoring. Hence none of those rather obvious techniques used for process monitoring can be applied to monitoring of sensors due to the 'unknown' nature of the input signal to the sensor system. Therefore, new techniques for sensor evaluation have to be considered. This key problem of unknown inputs imposes a dilemma to sensor monitoring which has to be resolved before one can monitor the sensor system. Without knowing the input, how can one implement a sensor monitoring scheme?

There are many techniques available for process and controller monitoring. There are

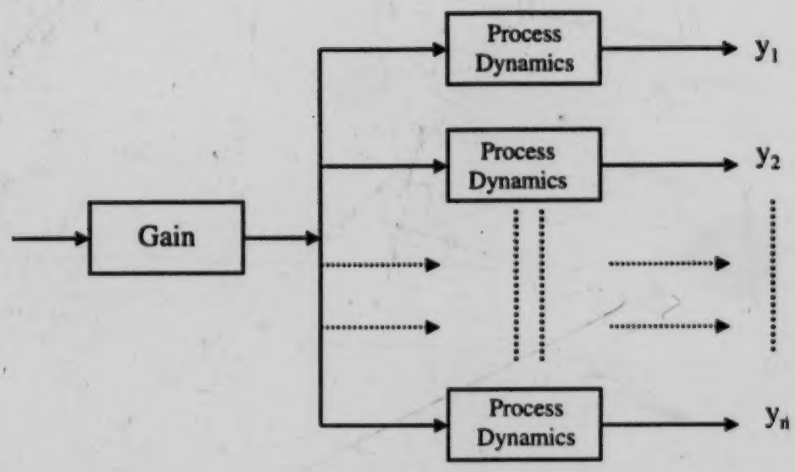

Figure 4.2: Combined Gain Monitoring

also many techniques available for sensor monitoring through the use of indirect input information. Many existing sensor monitoring schemes use input information indirectly from upstream processes or loops. Thus, these methods heavily depend on the availability of the input information. There are relatively fewer methods available for sensor monitoring without the knowledge of the inputs. Some of the available techniques include the application of multiscale analysis and dynamic PCA (Rongfu *et al.* 1999). Here the output is broken down into different frequencies using wavelets. The low frequency content is assumed to be the process dynamics and the high frequency the noise. The middle frequency content is assumed to represent the sensor dynamics and is monitored for faults. In another application of PCA (Ming and Babu 2000) the noise in the output signal is monitored to check for sensor faults.

In this chapter (section 4.2) a sensor decalibration detection method (O'Reilly 1998) which uses the local approach will be investigated. This method uses ARIMA modeling, which can be used to describe a slow drifting time series. Hence this method is applicable to systems with slow changing inputs.

However, for a truly robust faulty sensor detection algorithm it is important that the detection algorithm should also be robust to fast changing inputs. So a robust algorithm for the purpose of sensor monitoring should be independent of all input changes, which can be either slow or fast.

In section 4.3 an alternate method is proposed which is applicable to sensor monitoring problems where the input is not known but we will focus on the gain monitoring problem. This algorithm uses results for state estimations which are independent of the unknown inputs. This method is robust to both fast and slow changes in the input as the estimation is independent of the inputs. Work has been done in the field of observer design in which knowledge of inputs is not known (Darouch 1997, Kitanidis 1987). In this chapter the observer design algorithms in the above mentioned references are combined with the local approach for fault detection (Benveniste *et al.* 1987, Basseville 1998, Zhang *et al.* 1998, O'Reilly 1998) to derive a sensor decalibration algorithm which is independent of inputs. The algorithm is implemented upon a simulation example and a pilot scale stirred tank process. The algorithm developed can be used to monitor the gain decalibration of the process as shown in figure 4.2. Here it can be seen that in the framework of this algorithm the actuator gain change, the process gain change and the sensor gain change may all be monitored without the knowledge of inputs.

4.2 Sensor decalibration monitoring for slow-changing inputs

4.2.1 Monitoring algorithm using local approach

In this section, we will focus on monitoring of sensor decalibration. However, the result can be applied to any gain monitoring with unknown inputs. A particular method proposed by O'Reilly (1998) for sensor decalibration which uses the local approach is presented here.

The sensor output y in this method is modeled using an ARIMA model :

$$y_k = \frac{C}{A\Delta}\epsilon_k \tag{4.1}$$

where, ϵ_k is a white noise series, k is the time instant, Δ is the difference operator which is equal to $1 - z^{-1}$, A and C are polynomials in the back shift operator. This model structure can be used to describe slow drifting time series. The slow drifting property typically takes into account the slow changes in the sensor input. Equation 4.1 can be rewritten in terms of the prediction error:

$$\epsilon_k = \frac{A\Delta}{C}y_k \tag{4.2}$$

Assuming linear decalibration, the prediction error is:

$$\epsilon_k = \frac{A\Delta}{C}(py_k) \tag{4.3}$$

where p is a variable scalar quantity, having a value of $p_0 = 1$ under normal operation and other values under faulty operation. This can be viewed as a decalibration factor or a representation of the sensor (or process) gain. Equation 4.3 can also be written as:

$$\epsilon_k = p\sum_{i=0}^{q} a_i[y_{k-i} - y_{k-i-1}] - \sum_{i=1}^{r} c_i\epsilon_{k-i} \tag{4.4}$$

where k is the time index such that $t = kT$ (T is the sampling time), q and r are the orders of A and C polynomials, and a_i and c_i are the coefficients of A and C polynomials. The primary residual as discussed in the previous chapters can be written as the product of ϵ_k and its derivative:

$$H(\theta_0, y_k) = (\frac{\partial \epsilon_k}{\partial \theta})^T \epsilon_k$$

(This can be derived from the identification objective function which minimizes the square of the prediction error ϵ_k). The derivative can be calculated from equation 4.3 as:

$$(\frac{\partial \epsilon_k}{\partial p}) = \sum_{i=0}^{n} a_i[y_{k-i} - y_{k-i-1}] - \sum_{i=1}^{r} c_i(\frac{\partial \epsilon_{k-i}}{\partial p}) \tag{4.5}$$

(in this case p is the θ which is being monitored with other parameters being assumed to be constant) or using the transfer function notation:

$$(\frac{\partial \epsilon_k}{\partial p}) = \frac{A\Delta}{C} y_k = (1/p)\epsilon_k \tag{4.6}$$

Thus, the primary residual in this case is given by:

$$H(\theta_0, y_k) = \epsilon_k^2 \tag{4.7}$$

where $\theta_0 = p_0 = 1$. The normalized residual is given by:

$$\xi_N(\theta_0) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} \epsilon_k^2 \tag{4.8}$$

The χ^2 test is given by:

$$\chi_n^2 = \xi_N^T(\theta_0)\hat{\Sigma}^{-1}(\theta_0)\xi_N(\theta_0) \tag{4.9}$$

This is calculated over the testing data for every N sampling instances. The n degrees of freedom in this case correspond to the dimension of the monitored vector p, which has dimension one. The calculated χ^2 test value is then compared to the threshold value, obtained from the χ^2 table. A calculated value greater than this threshold indicates a possible sensor decalibration or fault.

4.2.2 Results

The primary objective of this algorithm is to detect slow sensor decalibration which would not be detected easily by visual observation, using the Local Approach. Analysis is carried on various simulated and industrial data to gauge the potential of the method as an early indicator of decalibration and abrupt faults. A fault would be indicated if the calculated value persistently exceeds the threshold. Results on a simulation are given as follows:

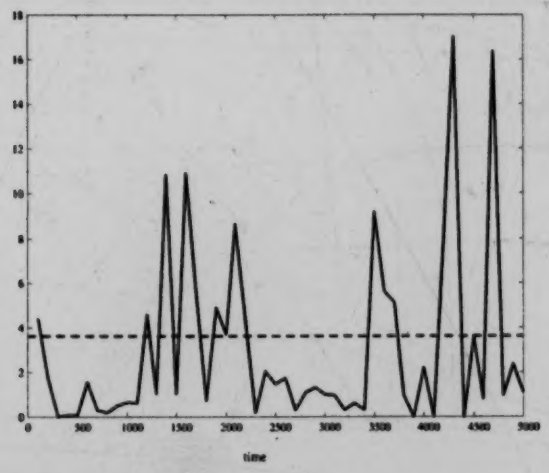

Figure 4.3: calculated χ^2 plot of a twenty percent change in the sensor gain at around one thousand sampling instances - the threshold is shown as a dotted line.

Example 4.1

The bio reactor given in section 2.4, equation 2.35, is taken as the example. Reproducing the model again:

$$\begin{aligned} \dot{x}_1 &= \frac{a x_1 x_2}{x_2 + b} - x_1 u \\ \dot{x}_2 &= \frac{d a x_1 x_2}{x_2 + b} + (c - x_2) u \\ y &= x_2 \end{aligned} \tag{4.10}$$

where

x_1: microbial concentration in the process,

x_2: substrate concentration in the process,

u: dilution rate,

a: maximum growth rate,

b: saturation parameter,

c: inlet substrate concentration,

d: yield factor.

The output from this process is assumed to pass through a sensor. The transfer function of this sensor is simply taken to be a gain of one. Input is taken to be a ramp function of slope 0.01. To simulate a decalibration the gain is changed. Results of two such simulations are shown in two figures. Figure 4.3 is the calculated χ^2 plot of a twenty percent change in the sensor gain at around one thousand sampling instances (a total of five thousand data points are tested). The fault is detected and can be seen as violation of the threshold. The threshold taken for this case is 3.84, with five percent allowable false alarm rate. Figure 4.4

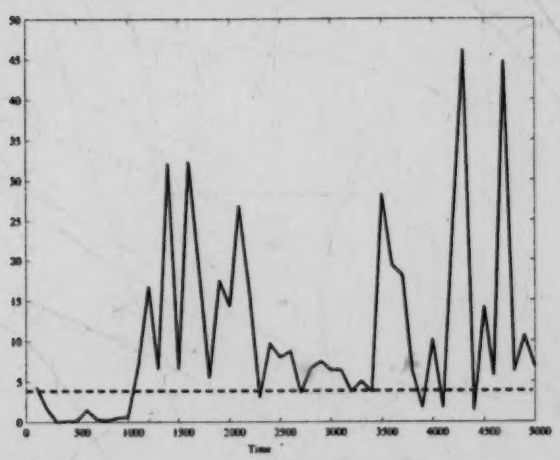

Figure 4.4: calculated χ^2 plot of a forty percent change in the sensor gain at around one thousand sampling instances - the threshold is shown as a dotted line.

is the calculated χ^2 plot of a forty percent change in the sensor gain at around one thousand sampling instances. Here the fault is detected clearly.

Test results on an industrial case study are shown in figure 4.5. The figure on the left gives the actual pressure of a hydrogen stream. The χ^2 plot of figure 4.5 (plot on the right) shows the calculated χ^2 values and the threshold plotted against time. A possible decalibration around sampling time 1500 to 4000 is indicated. These are in fact rapid changes in the input (in this case step changes by plant engineers). Another possible fault indication is around 14000 sampling instant which on careful inspection reveals a change in frequency of oscillation of the data series. It is observed that false alarms result due to rapid change of the input (the step changes). Thus this algorithm is ideal for monitoring of processes with slow changing inputs and has to be modified for processes with fast changing inputs.

4.3 Sensor decalibration monitoring for fast-changing input

4.3.1 Monitoring algorithm using input independent Kalman filtering

With the primary objective of developing an algorithm to detect sensor decalibration, a method is evolved which consists of using some results in input independent Kalman filter. The idea here is to model the sensor as a system which gets unknown inputs (Figure 4.1) and to devise a fault detection strategy for the system without knowledge of the inputs. A gain change would signify a fault in the sensor. Monitoring of this gain change is of the interest in this section.

Some recent results by Darouach *et al* (1997) in the development of Kalman filter will

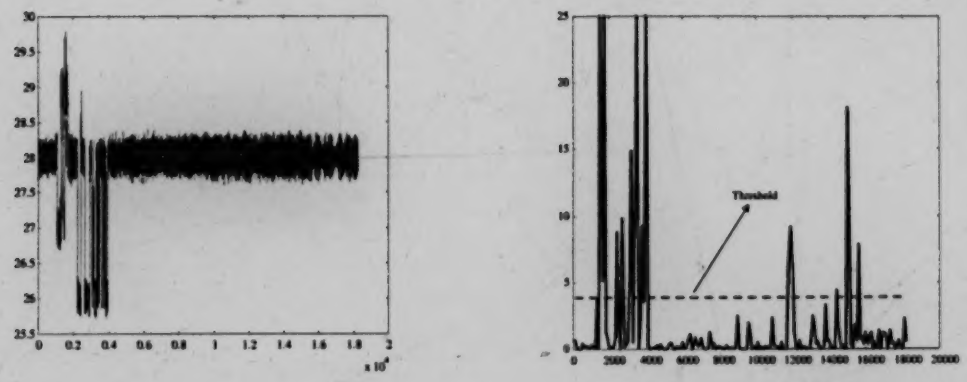

Figure 4.5: (left) time series plot of historized data and (right) the resulting χ^2 plot from the monitoring algorithm.

be used for this purpose. The method proposed by Darouach *et al.* (1997) to get unbiased minimum variance state estimations without the knowledge of inputs is presented in this section. The traditional approach for the Kalman filter design requires assumptions on the unknown inputs. There has been work done in this field by making assumptions on how the unknown input varies in time (Mehra 1970). In some cases the assumptions are on the nature of the unknown inputs such as assuming it to be a stochastic process with known mean and covariance or as a constant bias (Friedland 1969, Ignani 1990, Zhou and Zhang 1993). This new approach aims to relax the assumptions on the inputs and directly estimates the state without the knowledge of inputs.
The system structure assumed is:

$$\begin{aligned} x_{k+1} &= \Phi x_k + G_k d_k + w_k \\ y_k &= H_k x_k + v_k \end{aligned} \tag{4.11}$$

where $x_k \in \Re^n$ is the state, $d_k \in \Re^p$ is the unknown input, and $y_k \in \Re^m$ is the output. v_k and w_k are the zero mean, white noise sequences which are uncorrelated and have non-negative definite covariances R_k and Q_k respectively. Φ_k, G_k and H_k are known matrices of appropriate dimensions such that rank(H_k) = m and rank(G_k) = p, (m $\geq$ p). The method proposed by Darouach *et al.* (1997) uses the filter formulation as proposed by Kitanidis (1987). This linear filter for minimum variance unbiased estimation is given by equation 4.12.

$$\hat{x}_{k+1/k+1} = \Phi \hat{x}_{k/k} + L_{k+1}(y_{k+1} - H_{k+1}\Phi_k \hat{x}_{k/k}) \tag{4.12}$$

Remark 4 *The linear filter given in equation 4.12 can be derived from the more general form of a filter where we have no knowledge of the input* $\hat{x}_{k+1/k+1} = A^{(1)}_{t+1}\hat{x}_{k/k} + A^{(2)}_{t+1}$. *For unbiasedness it can be shown that we would get the requirement* $A^{(1)}_{t+1} - \Phi + A^{(2)}_{t+1}H_{t+1}\Phi = 0$. *Setting* $A^{(2)} = L_{k+1}$ *and* $A^{(1)} = \Phi - L_{k+1}H_{k+1}\Phi$, *we would obtain equation 4.12 (Kitanidis 1987).*

Since this filter (equation 4.12) must be unbiased the following constraint equation must hold for the filter:

$$E[\hat{x}_{k+1/k+1} - x_{k+1}] = 0 \tag{4.13}$$

By using equation 4.11 and 4.12, equation 4.13 can be written as:

$$L_{k+1}H_{k+1}G_k - G_k = 0 \tag{4.14}$$

The covariance matrix for the prediction error is given as:

$$P_{k+1} = E[(\hat{x}_{k+1/k+1} - x_{k+1})(\hat{x}_{k+1/k+1} - x_{k+1})^T] \tag{4.15}$$

By using equation 4.11 and 4.12, equation 4.15 can be written as:

$$P_{k+1} = (I - L_{k+1}H_{k+1})(\Phi_k P_k \Phi_k + Q_k)(I - L_{k+1}H_{k+1})^T + L_{k+1}R_{k+1}L^T_{k+1} \tag{4.16}$$

The method proposed by Darouach *et al.* (1997) to get the gain 'L_{k+1}' consists of parameterization of the solution of equation 4.14 in the form of equation 4.17 and then deriving the minimum variance estimation using the parameterization of 'L_{k+1}'. This method has been presented here in a compact form.

Equation 4.14 has a solution if rank($H_{k+1}G_k$)= rank(G_k)=p. The solution is given by equation 4.17:

$$L_{k+1} = G_k\Pi_k + K_{k+1}T_k \tag{4.17}$$

with $\Pi_k = (H_{k+1}G_k)^+ \in \Re^{p\times m}$ and $T_k = \alpha_k(I - (H_{k+1}G_k)(H_{k+1}G_k)^+) \in \Re^{(m-p)\times m}$, where α_k is an arbitrary matrix which must be chosen such that T_k is a full-row rank matrix and the matrix

$$\begin{bmatrix} T_k \\ \Pi_k \end{bmatrix} \tag{4.18}$$

is nonsingular. A^+ represents the pseudo-inverse matrix of A. Substituting this parameterization (equation 4.17) in the variance equation 4.16. The minimum variance estimate problem is reduced to finding the matrix K_{k+1} (refer equation 4.17) which minimizes the trace of the covariance matrix (equation 4.16).

Darouach *et al.* (1997) have given the following expression for estimating the states for a time invariant system:

$$\begin{aligned}\hat{x}_{k+1} &= F\hat{x}_k + G\Pi y_{k+1} + K_{k+1}(Ty_{k+1} - \beta\hat{x}_k)\\ \text{where:}\\ K_{k+1} &= (FP_k\beta^T + S)(\beta P_k\beta^T + \Theta)^{-1}\\ P_{k+1} &= (F - K_{k+1}\beta)P_k(F - K_{k+1}\beta)^T + Q_1 + K_{k+1}\Theta K_{k+1}^T - K_{k+1}S^T - SK_{k+1}^T\end{aligned} \tag{4.19}$$

and:

$$\begin{aligned}Q_1 &= \Sigma Q\Sigma^T + G\Pi R\Pi^T G^T\\ \Sigma &= I - G\Pi H\\ F &= \Sigma\Phi\\ \beta &= TH\Phi\\ S &= \Sigma QH^T T^T - G\Pi RT^T\\ \Theta &= T(HQH^T + R)T^T\\ \Pi &= (HG)^+\\ T &= \alpha(I - (HG)(HG)^+)\end{aligned} \tag{4.20}$$

The state estimate can be used to find the prediction errors or residuals:

$$\begin{aligned}\hat{y}_k &= H\hat{x}_k\\ \epsilon_k &= \hat{y}_k - y_k\end{aligned} \tag{4.21}$$

Using the earlier result on sensor decalibration (refer section 4.2), the primary residual can be defined as the square of the prediction residuals:

$$H(\theta_0, y_k) = \epsilon_k^2 \tag{4.22}$$

Recall that in the above derivation, an unbiased state estimation is guaranteed. Therefore, the prediction residual should have a zero mean. However, unlike the traditional Kalman filter, the residual obtained here may not be white. To obtain the true whiteness, we may need to cascade another whitening filter to the residual ϵ_k. This can easily be done by fitting ϵ_k with a time series model.

Limitations on the result

1. The solution of equation 4.14 exists if rank(HG) = rank(G). Also it is assumed rank(H) $\geq$ rank(G) and both H and G be full column rank matrices.

2. If HG is invertible, then there is only one solution. In that case equation 4.14 will yield a unique solution.

3. Under condition 1, multiple solutions exist only if HG is not a square matrix.

4. For a single input single output system, there is a unique solution. Therefore, for multiple solutions to exist, the system must be at least multivariate in the output.

Other limitations and conditions on this method along with conditions for the detectability and reachability of the states are discussed in the work by Darouach *et al* (1997) in detail. The stability and convergence conditions for this filter are presented in their work. It consists of deriving the algebraic Riccati equation (ARE) corresponding to the system and then finding the stability conditions based on this ARE. To derive these conditions the authors have utilized the results for the stability and convergence of standard Kalman filter (Anderson and J.B.Moore 1989, Bitmead *et al.* 1990).

4.3.2 Simulation and pilot plant case study

For the simulation a two input two output system is used, and for the experiment a system with one input (cold water) and two outputs (level and temperature) is used. The pilot plant in the experiment also has a steam input to the coil, but that is kept constant throughout the experiment so its effect can be neglected.

Simulation

Example 4.2

For the simulation a two-input two-output system is taken:

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.9512 & 0 & 0 & 0 \\ 0 & 0.9048 & 0.0669 & 0.0227 \\ 0 & 0 & 0.8825 & 0 \\ 0 & 0 & 0 & 0.9048 \end{bmatrix} x_k + \begin{bmatrix} 0.4877 & 0.4877 \\ -1.1895 & 3.5690 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u_k + w_k \\ y_k &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \hat{x}_k + v_k \end{aligned} \tag{4.23}$$

where v_k and w_k are white noise with covariance matrices Q and R respectively given by:

$$\begin{aligned} Q &= 10^{-2} \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.004886 & 0 & 0 \\ 0 & 0 & 0.2212 & 0 \\ 0 & 0 & 0 & 0.7252 \end{bmatrix} \\ R &= 10^{-5} \times \begin{bmatrix} 0.1 & 0 \\ 0 & 0.6 \end{bmatrix} \end{aligned}$$

Table 4.1: *Results of sensor decalibration algorithm on simulated data.*

no change	10% change	20 % change	30 % change
0.0032	17.8857	63.1454	187.2420

It is assumed that the output y_k passes through a sensor. It is also assumed in the following fault detection simulation, that the input u_k is unknown. This state space model (equation 4.23) is used to generate the outputs of the system. The input is taken as a random number with zero mean and unit variance, the noise terms are also taken as random numbers with zero means and variance taken from the respective covariance matrices. No plant-model mismatch is assumed and the same model is used for the design of the input independent Kalman filter. The input independent Kalman filter based sensor decalibration algorithm is run on the data generated. In this case, a unique solution of the input-independent Kalman filter gain exists as HG is invertible. The sensor fault or decalibration is generated by changing the sensor gain. The results over hundred simulation runs using different noise seeds are shown in table 4.1. Here the calculated χ^2 is shown for each case of ten, twenty and thirty percent change in the sensor gain. The threshold in this case is 3.84 with a five percent false alarm rate.

The simulation results show that the algorithm is able to detect the simulated sensor fault, i.e. the change in the sensor gain. This is shown by the violation of the threshold by the calculated χ^2 value.

Pilot plant case study

The pilot plant, a schematic of which is shown in figure 4.6, in the Computer Process Control laboratory at the Department of Chemical and Materials engineering, University of Alberta is considered.

The set up shown in figure 4.6 is a tank with two inputs and two outputs. Cold water flows continuously into the tank, and the level of water in the tank is measured by a level transmitter (shown as LT in the figure). The water is heated by means of steam passing through a heating coil in the tank. The outlet temperature of the water is measured by a thermocouple in the outlet pipe from the tank, as shown in the schematic diagram (shown as TT1 to TT4 in the figure). The flow of water and steam to the tank can be varied by control valves on the inlet cold water and steam lines. The system is interfaced to a computer through a real time Simulink/Matlab interface. For our analysis a single input multiple output arrangement is taken as a non unique solution is desired (refer to section 4.3.1). The single input is the cold water flow and the steam inlet flow is kept constant. The level of the tank and the temperature of the outlet (measured by the first downstream thermocouple TT1) are considered to be the two outputs of the process. The

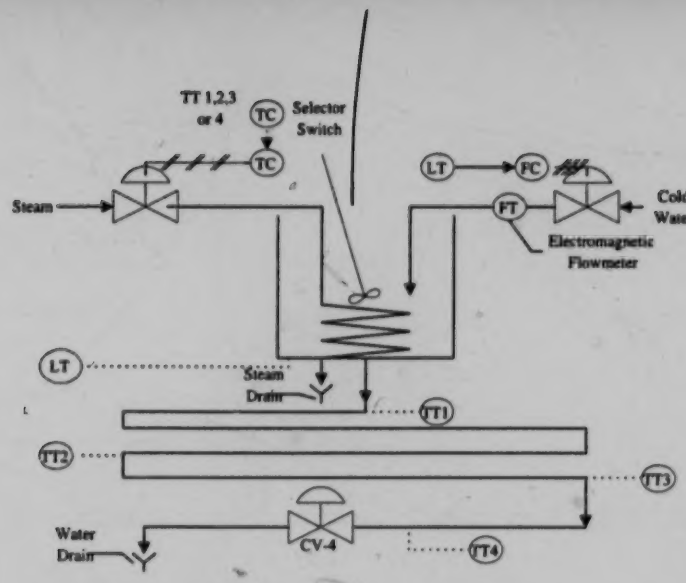

Figure 4.6: schematic of the pilot plant set up

sampling time is taken to be five seconds. A state space model identified using persistently excited input signal is given by:

$$\begin{aligned} \hat{x}_{k+1} &= \begin{bmatrix} 0.9103 & -0.0383 & -0.0486 & -0.0016 \\ -0.5879 & 0.4319 & -0.1669 & 0.1842 \\ -0.6523 & -0.2223 & 0.1922 & -0.0801 \\ -0.4360 & 0.0360 & -0.7125 & -0.0896 \end{bmatrix} \hat{x}_k + \begin{bmatrix} 0.0541 \\ -0.0428 \\ 0.5541 \\ 0.8699 \end{bmatrix} u_k + w_k \\ y_k &= \begin{bmatrix} 0.9275 & -0.0568 & -0.0358 & -0.0212 \\ -0.0563 & -0.0105 & 0.0031 & 0.0010 \end{bmatrix} \hat{x}_k + v_k \end{aligned} \tag{4.24}$$

where v_k and w_k are white noise with covariance matrices Q and R respectively given by:

$$\begin{aligned} Q &= \begin{bmatrix} 0.0001 & -0.0006 & -0.0006 & 0.0016 \\ -0.0006 & 0.0085 & 0.0096 & -0.0126 \\ -0.0006 & 0.0096 & 0.0130 & -0.0151 \\ 0.0016 & -0.0126 & -0.0151 & 0.0252 \end{bmatrix} \\ R &= 10^{-3} \times \begin{bmatrix} 0.1058 & -0.0359 \\ -0.0359 & 0.1746 \end{bmatrix} \end{aligned} \tag{4.25}$$

The input independent Kalman filter based decalibration detection algorithm is run on two different set of data, first over the identification data which is persistently excited and then over a routine operation data, i.e. data around the steady state with few step changes. The cold water input is assumed to be unknown in both cases for the design of the Kalman filter. A sensor fault or decalibration is visualized as a simple change of the gain of the process. The outputs for the identification data set for no change in the sensor gain are

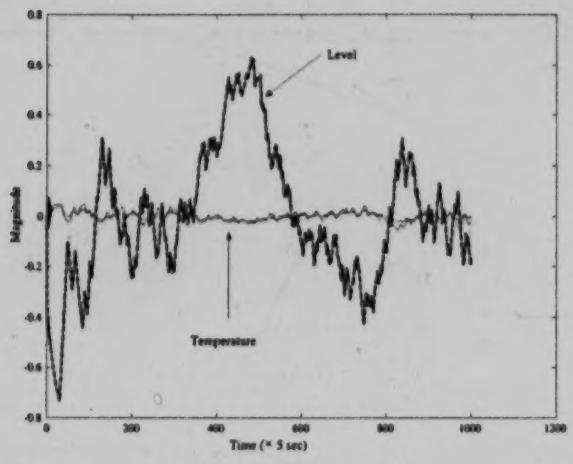

Figure 4.7: plot of the outputs along with the estimates from the input independent Kalman filter. The solid line corresponds to actual outputs and the dotted line the estimated outputs.

given in figure 4.7. It is seen here that a good estimate of the outputs is obtained without the knowledge of the inputs.

Ten, twenty and thirty percent change in the sensor gain is generated at the two hundredth sampling instance. A representative plot for actual outputs with no change and output with twenty percent change in gain at the two hundredth sampling instance is shown in figure 4.8. The estimate for the twenty percent gain change along with the actual outputs are shown in figure 4.9.

Similarly the analysis is carried out for a batch of routine data. The input in this case is a few step changes. The output and estimate plots for no sensor fault are given in figure 4.10. As in the analysis with the identification data ten, twenty and thirty percent change in the sensor gain is generated at the two hundredth sampling instance. A representative plot for actual outputs with no change and output with twenty percent change in gain at the two hundredth sampling instance is shown in figure 4.11. The estimate for the twenty percent gain change along with the actual outputs are shown in figure 4.12.

In each case before the detection algorithm is tested on these two data sets the prediction residuals are pre-whitened to ensure that the residuals being tested are white noise. For this purpose the first three hundred of the prediction residuals are trained by a second order AR model and then all the residuals are filtered by this AR model to get the whitened residuals. The primary residuals are obtained from these whitened residuals. The detection algorithm is run over the data taking ten, twenty and thirty percent change in the sensor gain over the two data sets. The χ^2 values for each case is given in table 4.2. The threshold in this case is 3.84 with a five percent false alarm rate.

The algorithm is found to be sensitive to sensor gain change while being independent of

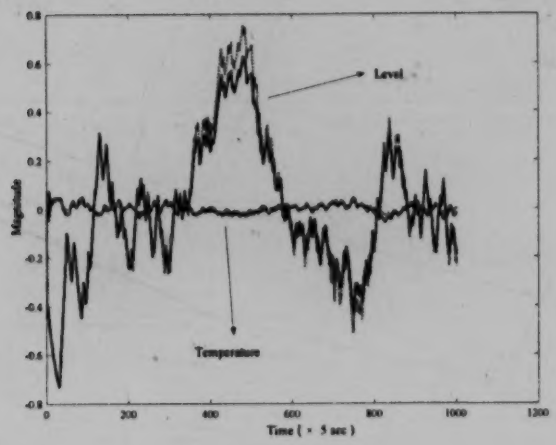

Figure 4.8: plot of the actual outputs (solid line) along with the outputs with 20 % gain change (dotted line) at the two hundredth sampling instance.

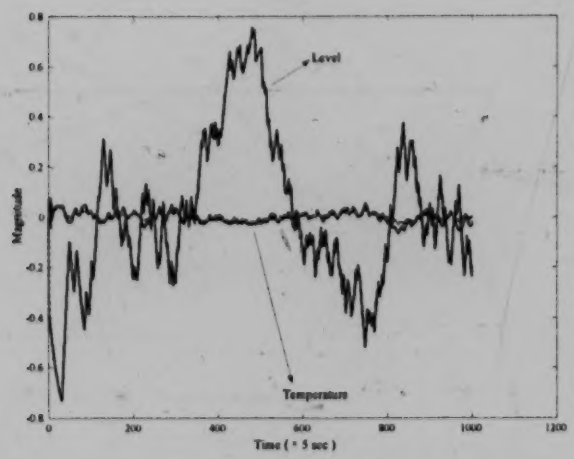

Figure 4.9: plot of the outputs with a 20 % change in the sensor gain at the two hundredth sampling instance (solid line) along with the estimates from the input independent Kalman filter (dotted line).

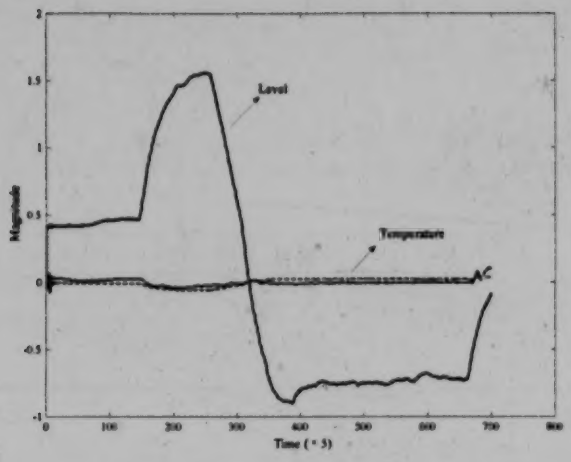

Figure 4.10: plot of the outputs (solid line) along with the estimates from the input independent Kalman filter (dotted line) for routine data.

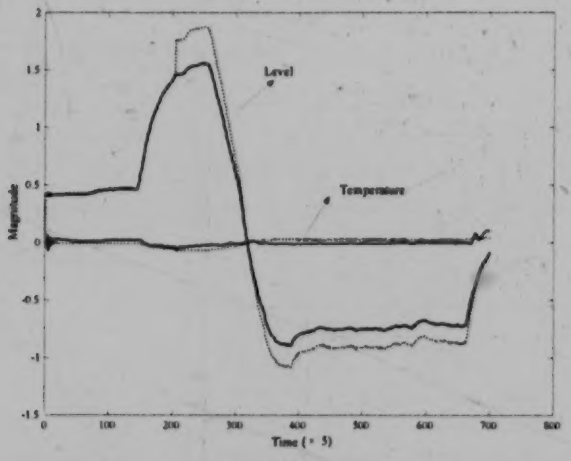

Figure 4.11: plot of the outputs (solid line) along with the outputs with 20 % change in the sensor gain at the two hundredth sampling instance (dotted line).

Table 4.2: *Results of Sensor decalibration algorithm with different changes in the sensor gain at the two hundredth sampling instance*

	no change	10% change	20 % change	30 % change
identification data	0.0042	14.0730	61.8706	152.8949
routine data	0.0023	0.8169	7.4957	30.3215

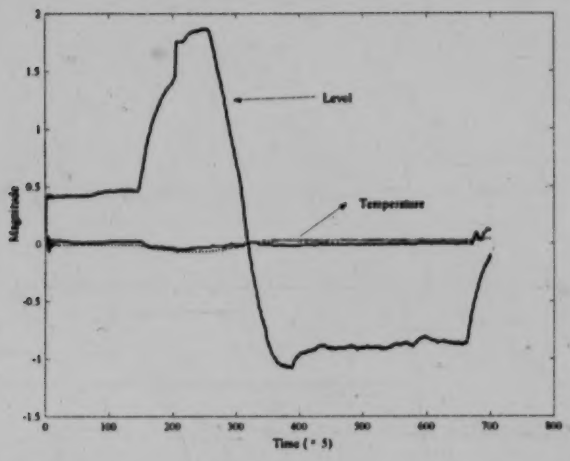

Figure 4.12: plot of the outputs (solid line) with a 20 % change in the sensor gain at the two hundredth sampling instance along with the estimates from the input independent Kalman filter (dotted line).

the changes in inputs.

4.4 Conclusions

This chapter focuses on fault detection and process monitoring for a class of systems where the input is not known. We have focused on the case of sensor monitoring since the inputs to a sensor are not known. An algorithm for detection of sensor decalibration which uses the local approach with ARIMA modeling is studied in this chapter. The algorithm gives good results for a system with slow changing inputs which is as expected since the ARIMA modeling can handle slow drifting time series. The drawback is however that fast input changes may also be reflected as sensor faults as seen in the industrial application where step changes show up as faults (figure 4.5). The issue of devising a sensor decalibration strategy which is robust to both slow and fast input changes has been addressed using the input independent Kalman filter based detection algorithm.

The detection algorithm based on input independent kalman filtering and local approach is found to be sensitive to the change in sensor (or process) gain and independent to all changes in the input, as observed from the simulation and pilot plant case study results. This algorithm can be used to monitor the change in process gain in general for cases where the inputs are now known. In particular it can be used to monitor the decalibration of sensors. The drawback of the algorithm is lack of an isolation methodology at this point to isolate which sensor or actuator is at fault.

Chapter 5

Summary and Conclusions

A summary of the major contributions of this thesis is as follows:

Implementation of fault detection approaches

This work deals with the application of fault detection approaches to chemical process industries. Special emphasis has been laid on the implementation of the statistical local approach (Benveniste *et al.* 1987) to fault detection and model validation. This scheme is tested on simulation examples and industrial processes. The results from these simulations and the industrial examples show promise and potential in early detection of undesirable changes in processes. Without a monitoring strategy these faults may escape detection till it is too late and may have a devastating affect on the process. The local approach based monitoring scheme shows comparable or better results than standard techniques like PCA and SVD based monitoring schemes.

Integration of the approach with identification strategies has been exploited in this work. This approach has been applied in synergy with various model formulations like AR, PEM, OE. The local approach detection algorithm is successfully applied for fault detection on industrial data of a distillation column and a fluidized coker. The industrial examples, especially those done for the monitoring of the industrial distillation column show that the algorithm is suitable for on-line early detection of process faults. It is seen that the local approach based detection algorithm is able to detect the flooding of the distillation column in question, around three hundred sampling instances before it becomes apparent to the plant personnel. Thus this scheme can be used for early detection of process faults and deterioration of the system performance. Also for the distillation column monitoring it is seen that the best results are obtained when the model is based on the first-principle mass balance equation. This shows that the performance of fault detection can be improved if apriori information is incorporated into the detection algorithm (e.g. the mass balance equation) instead of just using black box modeling (SVD or regression modeling). For the industrial data it is also observed that fine-tuning of the thresholds is required, because the thresholds got from tables or relations may be too tight as far as actual process is concerned

and may have to be relaxed to get rid of excessive false alarms. This monitoring scheme is tested successfully online on an industrial process simulator at Syncrude Canada Upgrading facility at Ft. McMurray.

Control relevant model validation

The local approach is used in synergy with a control relevant model identification strategy which uses data prefiltering to derive a control relevant model validation strategy. The algorithm is derived with the objective of validating models for model predictive control. Traditional residual tests for validation of models check all changes even if the change may not affect the performance of the controller which relies on the model for predictions. Hence such a validation test based on residual analysis may be too restrictive or too loose a conditions to impose on a model used for predictive control. The new algorithm developed for control relevant validation is sensitive to only those changes that are most likely to affect the controller performance. This strategy attempts to *'see'* the model through the *'eye'* of the controller. Hence this validation strategy shows a major improvement over the regular residual whiteness test as far as validating models for advanced control applications are concerned.

This algorithm has scope for wide spread industrial application due to the recent increase in the number of advanced control applications in the chemical process industry. It can prove to be a useful tool in assessing and improving the performance of the advanced controller. This presents a new angle for improving the predictive controller performance by giving the engineer the tool to check if the model used is suitable for the task or not. The simulation and industrial examples (on the fluidized coker and the hydrogen plant data) show that it may not be wise to discard a model without considering the intended usage of the model, i.e. careful consideration is required to test the model over the prediction horizons over which the controller is operating. This algorithm can be used on multivariate systems by breaking them down into MISO systems and then validating each MISO system individually. For MISO systems isolation is also incorporated in the algorithm to pinpoint the particular model which is at fault.

Gain monitoring of processes without the knowledge of inputs with specific attention to the case of sensor monitoring

Last, the problem of fault detection and monitoring of processes without the knowledge of inputs is addressed. Sensor monitoring falls in this category of detection problems. This is because for a typical sensor only outputs are known (which are the process measurements) and the inputs to the sensor are really unknown. Sensor decalibration is a very common yet often ignored problem in chemical industries. This is because there is a dearth of reliable tools for sensor monitoring, mainly due to the issue of the inputs to the sensor being unknown, although there is a large volume of papers concerned with sensor monitoring

through indirect input information (if it is available).

An existing method in the literature which uses the local approach along with ARIMA modeling for testing sensor decalibration is analyzed (O'Reilly 1998). It is found that the algorithm is able to detect the decalibration of the sensor, but it is sensitive to fast changes in the process inputs and may give false alarms. To tackle this problem a new method is proposed which uses input independent Kalman filtering to give unbiased minimum variance state estimates (Darouch 1997). The algorithm derived is effective in monitoring sensor decalibration. It can also be used in the more general case of process gain monitoring where the inputs are unknown. This aspect of the approach has practical application for process gain monitoring in the chemical process industry where some input streams like recycle and gas inlet streams may be unknown or the input sensor reading may be unreliable. The algorithm is tested successfully on a simulation example and on a pilot plant experiment. There is also scope of future work in this field. The algorithm can be extended to include the isolation of the particular faulty sensor.

All algorithm are coded in Matlab and a graphical user interface is made for the control relevant model validation (Appendix A).

Bibliography

Abhorey, S. and O. Williamson (1978). State and parameter estimation of microbial growth processes. *Automatica* **14, No.5**, 493–498.

Anderson, B.D.O. and J.B.Moore (1989). *Optimal Filtering.* Prentice Hall. Englewood Cliffs, NJ.

Apley, D.W., Shi J. (1994). A fault detection, isolation and identification technique for complex MISO linear systems. *Proceedings of the American Control Conference* **FA 10-10:40**, 2633–2637.

Aravena, J.L., Chowdhury F.N. (1996). New approach to fast fault detection in power systems. *Proceedings of the International Conference on Intelligent Systems Applications to Power Systems, ISAP* pp. 328–332.

Basseville, M. (1983). Sequential detection of abrupt change in spectral characteristics of digital signals. *Automatica* **AC-29, No.5**, 709–724.

Basseville, M. (1988). Detecting changes in signals and systems - a survey. *Automatica* **24(3)**, 309–326.

Basseville, M. (1998). On-board component fault detection and isolation using the statistical local approach. *Automatica* **34**(11), 1391–1415.

Basseville, M. and I.V. Nikiforov (1993). *Detection of Abrupt Changes.* Prentice Hall. Englewood Cliffs, New Jersey.

Benveniste, A., Basseville M. and Moustakides (1987). The asymptotic local approach to change detection and model validation. *IEEE Trans. On Automatic Control* **AC-32, No.7**, 583–592.

Bitmead, R.R., M. Gevers and V. Wertz (1990). *Adaptive Optimal Control.* Prentice Hall.

Camacho, E.F. and C. Bordons (1998). *Model Predictive Control.* Springer Verlag.

Darouch, M., Zasadzinski M. (1997). Unbiased minimum variance estimation for systems with unknown exogenous inputs. *Automatica* **33, No.4**, 717–719.

Frank, P.M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge based redundancy - a survey and new results. *Automatica* **26**, 459–474.

Friedland, B. (1969). Treatment of bias in recursive filtering. *IEEE Trans. Autom. Control* **AC-14**, 359–367.

Gertler, J.J. (1988). Survey of model based failure detection and isolation in complex plants. *IEEE Control System Mag.* **8(6)**, 3–11.

Hall, W.J. and D.J. Mathiason (1990). On large sample estimation and testing in parametric models. *International Statistical Review* **58, No.1**, 77–79.

Hjalmarsson, H., M. Gevers, F.D. Bruyne and J. Leblond (1994). Identification for control: Closing the loop gives more accurate controllers. In: *Proceedings of 1994 CDC*. Lake Buena Vista, FL. pp. 4150–4155.

Huang, B. (1999*a*). Detection of abrupt changes of total least squares models and application in fault detection. *To appear in IEEE Trans. On Control System Technology.*

Huang, B. (1999*b*). System identification based on last principal components analysis. In: *Proceedings of IFAC'99 World Congress.* Beijing, China. pp. 211–216.

Huang, B. (2000). Multivariable model validation in the presence of time-variant disturbance dynamics. *Chemical Engineering Science, to appear.*

Huang, B. and A. Malhotra (2000). MPC relevant model identification and validation. *Internal Report.*

Huang, B. and S.L. Shah (1997). Closed-loop identification: a two-step approach. *Journal of Process Control.*

Ignani, M.B. (1990). Separate bias Kalman estimator with bias state noise. *IEEE Trans. Autom. Control* **AC-35**, 338–341.

Isermann, R. (1984). Process fault detection based on modeling and estimation methods - a survey. *Automatica* **20**, 387–404.

Kitanidis, P. K. (1987). Unbiased minimum variance linear state estimation. *Automatica* **23, No.6**, 775–778.

Klema, V.C. and A.J. Laub (1980). The singular value decomposition: Its computation and some applications. *IEEE Trans. on Automat. Control* **AC-25, No.2**, 164 – 176.

Lakshminarayan, S. (1997). Process characterization and control using multivariate statistical techniques. PhD thesis. University of Alberta.

LeCam, L. (1986). *Asymptotic Methods in Statistical Decision Theory.* Springer,New York.

Ljung, L. (1997). *System Identification.* 2nd ed.. Prentice-Hall.

Ljung, L. and T. Glad (1994). On global identifiability for arbitrary model parametrizations. *Automatica* **30, No.2**, 265–276.

Marcu, T., Mirea L. (1994). Robust detection and isolation of process faults using neural networks. *IEEE Control Systems Magazine* **17,Oct. 17 No. 5**, 72–79.

Maryak, J.L., L.W. Hunter and S. Favin (1997). Automated system monitoring and diagnosis via singular value decomposition. *Automatica* **33**(11), 2059–2063.

Mehra, R.K. (1970). On the identification of variances and adaptive Kalman filtering. *IEEE Trans. Autom. Control* **AC-15**, 175–184.

Ming, Y.C. and J. Babu (2000). Sensor fault detection using noise analysis. *Industrial and Engineering Chemistry Research* **39, No. 2**, 396–407.

Monad, J. (1950). La technique de cultures continues: théorie et applications.. *Ann. Inst. Pasteur* **79, No.4**, (In french).

O'Reilly, P.G. (1998). Detection of sensor decalibration using the asymptotic local approach. *ELECTRONIC LETTERS* **34, No.21**, 2022–2023.

Patton, R.J., P. Frank and R. Clark eds. (1989). *Fault Diagnosis in Dynamic Systems-Theory and Application.* Prentice-Hall, Englewood Cliffs, NJ.

Ritt, J. (1950). *Differential Algebra.* American Mathematical Society, Providence, RI.

Rongfu, L., M. Mishra and D.M. Himmelblau (1999). Sensor fault detection via multiscale analysis and dynamic PCA. *Industrial and Engineering Chemistry Research* **39, No. 2**, 396–407.

Shook, D.S., C. Mohtadi and S.L. Shah (1992). A control-relevant identification strategy for GPC. *IEEE Trans. AC.*

Soderstrom, T. and P. Stoica (1989). *System Identification.* Prentice Hall International. UK.

Sternad, M. and T. Soderstrom (1988). Lqg-optimal feedforward regulators. *Automatica* **24**(4), 557–561.

Van den Hof, P.M.J. and R.J.P. Schrama (1995). Identification and control—closed-loop issues. *Automatica* **31**(12), 1751–1770.

Willsky, A.S. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica* **12**, 713–725.

Zang, Z., R.R. Bitmead and M. Gevers (1995). Iterative weighted least-squares identification and weighted LQG control design. *Automatica* **31**(11), 1577–1594.

Zhang, Q., M. Basseville and A Benveniste (1994). Early warning of slight changes in systems. *Automatica* **30**(1), 95–113.

Zhang, Q., M. Basseville and A Benveniste (1998). Fault detection and isolation in non-linear dynamic systems: A combined input-output and local approach. *Automatica* **34**(11), 1359–1373.

Zhou, D., Y.X. Sun Y.G. Xi and Z.J. Zhang (1993). Extension of friedland's seperate bias estimation to randomly time-variant bias of nonlinear systems. *IEEE Trans. Autom. Control* **AC-38**, 1270–1273.

Appendix A

Graphical User Interface for predictive control relevant model validation

The generic matlab GUI 'vgui' is the implementation of the control relevant model validation algorithm using the output error method (refer chapter 3). The parameters specified by the users in GUI are the number of manipulated variables (inputs to the systems), the number of disturbance variables (measured disturbances to the system also taken as inputs), the data matrix, and the models corresponding to the controlled variable (output) to be validated. The validation tests are performed on the model for prediction horizons from one to fifty. The output of the validation test is a plot of test score values (χ^2 values) plotted against the prediction horizon with a threshold value; any calculated χ^2 value which is above this threshold, implies that the model fails the validation test for that particlar prediction horizon. Consequently any calculated χ^2 value which is below the threshold implies that the model passes the test.

A.1 Getting started

The following guidelines should be followed to start this GUI.

1. Save the gui ('vgui') and all the other ('*.m' and '*.mat') files in the same folder or subdirectory.
2. Open the matlab workspace and make your workspace subdirectory or folder the same as the one in which the GUI is saved. The default directory when matlab is opened is usually '*drive* : *matlab**work*', this can be changed to the subdirectory of the GUI by using the matlab command 'cd'. For example if the gui is in 'c:' drive, 'user' subdirectory and 'gui' folder, in the matlab workspace type:
   cd *c* : *user**gui*
3. Open the GUI by typing the name of the GUI (i.e. 'vgui') in the matlab workspace once matlab working directory is the same directory as the GUI directory (or folder). The GUI figure will open (figure A.1) and the user can start using it.

A.2 Inputs to the GUI

The inputs to the GUI are labeled in figure A.1 and A.2. Figure A.1 shows the following inputs:

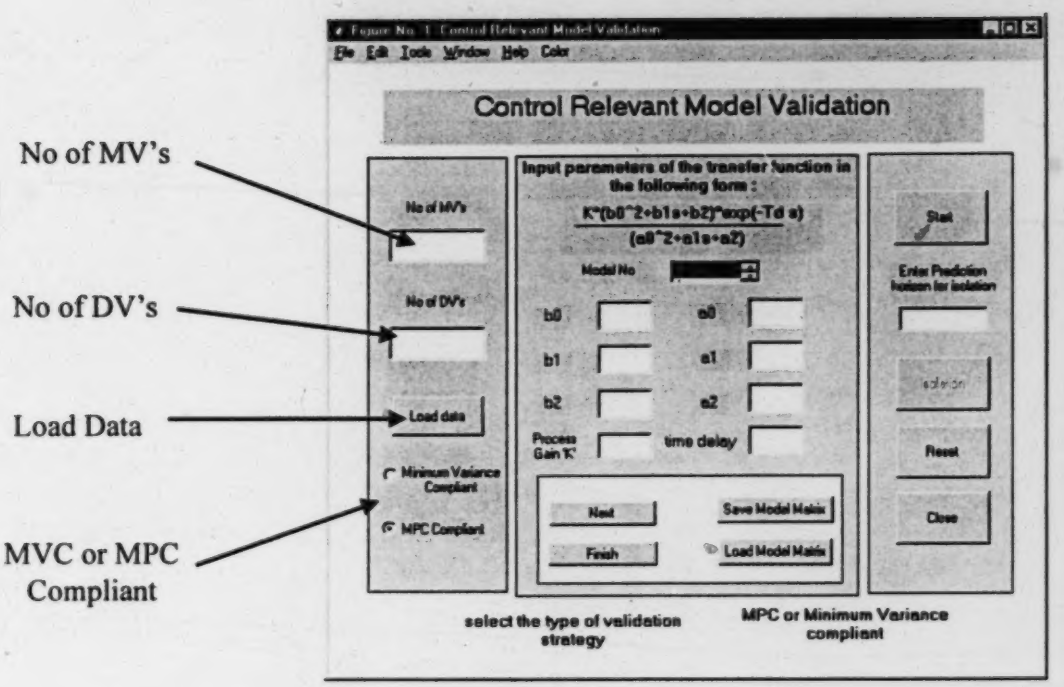

Figure A.1: Graphical User Interface, Inputs

1. **No of MV's:** The first box to the top-left is the box in which the number of Manipulated Variables is plugged in. A numeric quantity is expected which corresponds to all manipulated variables of the system in consideration.
2. **No of DV's:** The second box to the left is the box in which the number of Disturbance Variables is plugged in. A numeric quantity is expected which corresponds to the number of measured disturbance variables of the system in consideration.
3. **Data file:** The third box to the left is the box in which the data file (*.mat file) which has the data for the system being validated is expected; for example, 'datafile.mat'. The data stored in this file should be saved as '[Controlled Variable data, Manipulated Variable data, Disturbance Variable data]'.
4. **k step or multi-step option:** Check the circle with either k-step (Minimum Variance Control compliant) or multi-step prediction horizon method (Model Predictive Control compliant) option.

Figure A.2 gives the inputs which are needed to input the models which are to be validated. These buttons will only be enabled once all the information shown in figure A.2 is complete. The general form of the continuous model the GUI takes is:

$$y = K \times \frac{b_0 s^2 + b_1 s + b_2}{a_0 s^2 + a_1 s + a_2} e^{-T_d s}$$

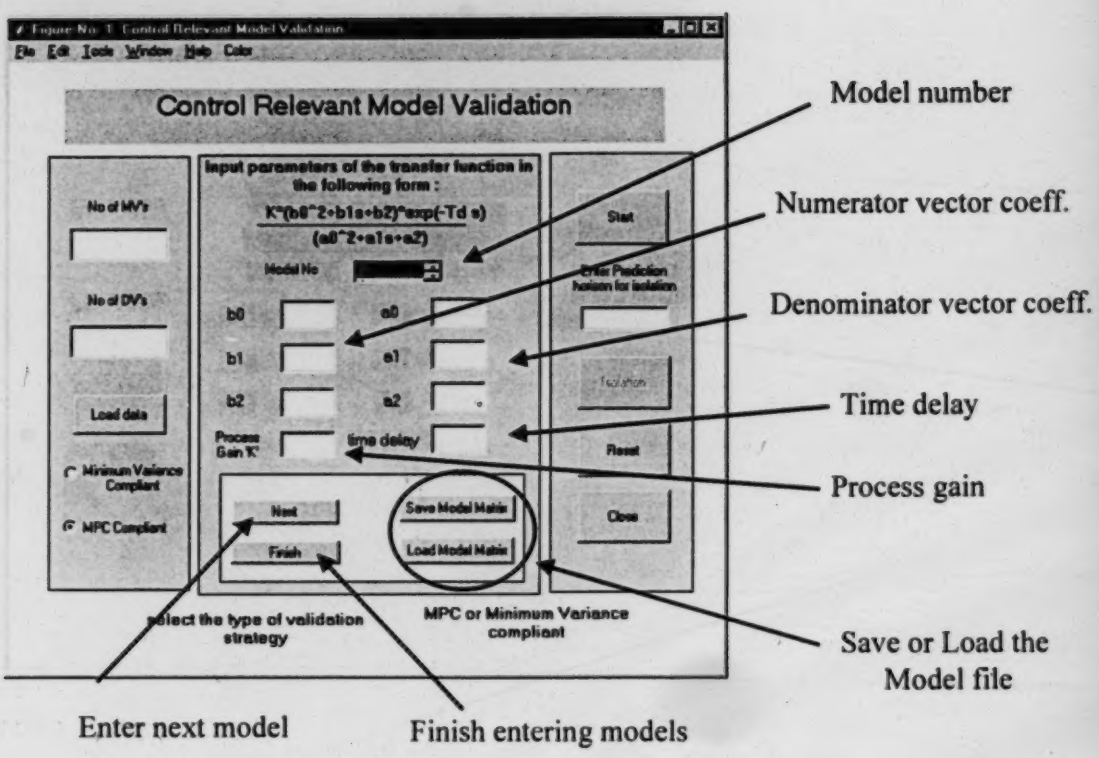

Figure A.2: Graphical User Interface, Inputing the models

1. **Model Number:** Each of the manipulated variables and the disturbance variables is considered to be an input. The convention for the numbering of the models in question is all the manipulated variables come first and the disturbance variables come next. For example if a particular output has five manipulated variables and nine controlled variables, the total inputs would be fourteen with inputs one to five being the manipulated variables and the inputs six to fourteen being the disturbance variables. This is a listbox which gives the option of selecting the input number corresponding to the model being fed to the validation algorithm.

2. **Numerator vector:** Once the input/model number has been fed, the next step is to feed the model to the algorithm. These boxes are for plugging in the numerator coefficient vector corresponding to the continuous model which links the particular input to the output. for example if our model is

$$1.5 \times \frac{0.8s^2 + 0.7}{s^2 + s + 1} e^{-2s}$$

the numerator vector would be b0 : 0.8, b1 : 0, b2 : 0.7.

3. **Denominator vector:** These boxes are for plugging in the numerator coefficient vector corresponding to the continuous model which links the particular input to the output. For the example considered above the denominator vector would be a0 : 1, a1 : 1, a2 : 1.

4. **Time Delay 'T_d':** This box is for specifying the time delay corresponding to that particular input. For the example considered above it would be 2.

5. **Process Gain 'K':** Plug in the process gain corresponding to the model here. For the model considered it would be 1.5.

6. **Next:** Once all the information for the input model is obtained from the Model number, Numerator and Denominator vectors and the time delay, we may want to input the next model. To let the algorithm take in the information already given and to input the next model this push button should be pressed after each model information is complete. This would clear the existing model information from the GUI boxes and the next model can be fed in (after assigning the correct model number).

7. **Finish:** If the model information just fed in is the last model (model corresponding to the last input) this push button should be pressed to give a signal to the algorithm that all the models have been given.

8. **Save Model Matrix:** Once all the models have been fed it is a good idea to save them in a 'mat' file so that we do not have to feed them every time we validate the same process. Pressing this pushbutton opens a dialogue box 'Save As' and the model matrix can be saved as a mat file in any user specified directory under a user specified name. The models would be stored under a matrix model in that mat file.

9. **Load Model Matrix:** If we already have the model matrix for a particular system saved through previous applications we do not have to feed in all the numerator, denominators etc all over again but can directly load back the file (should be a matlab 'mat file'). Pressing the pushbutton opens a matlab dialogue window 'Open Model .mat file' and the model file can be selected and loaded for validation (figure A.3).

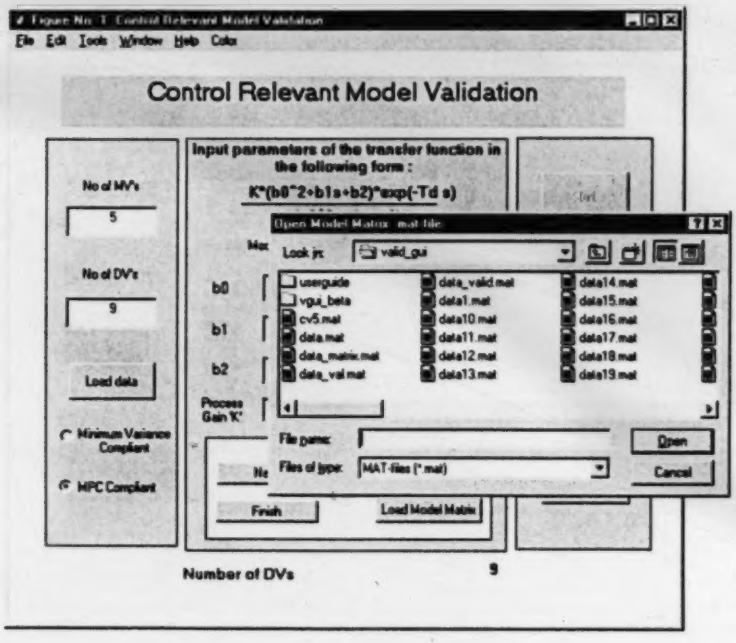

Figure A.3: load model file dialogue box

10. **Message display:** These message boxes will give helpful messages along each step of the GUI and can be used to see if the right contents have been fed into the GUI.

These are all the inputs needed to start our validation and isolation algorithms.

A.3 Outputs of the GUI

Figure A.4 gives all the output and other buttons and features of the GUI. These will be enabled once the inputs are complete.

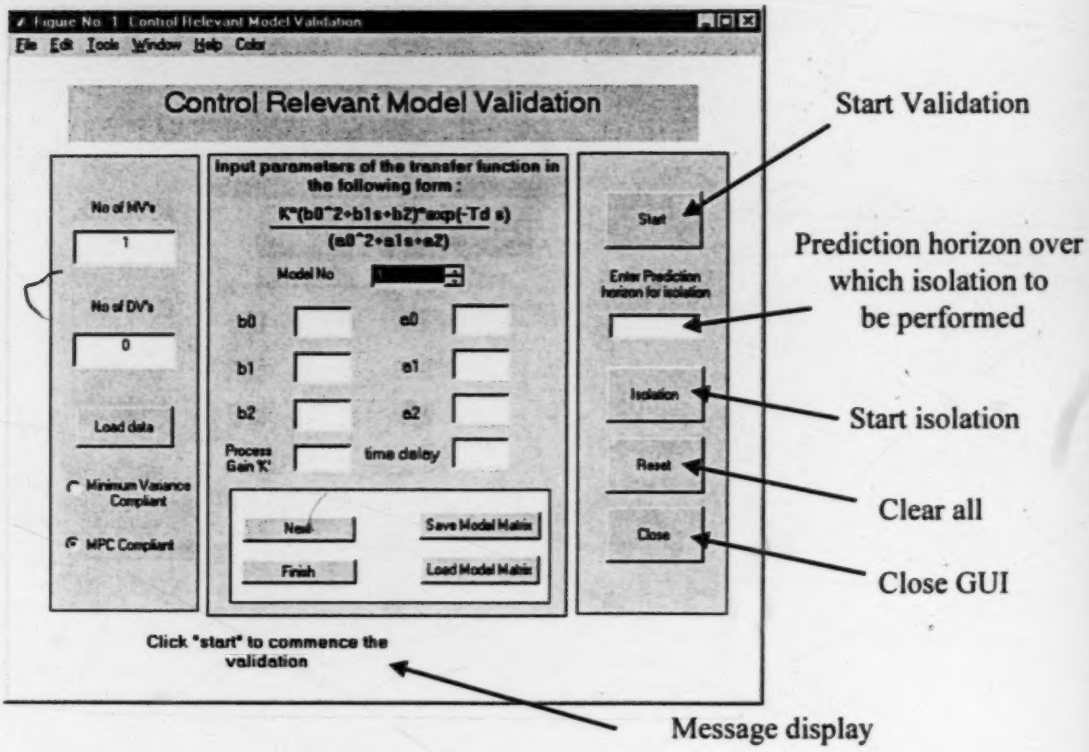

Figure A.4: Graphical User Interface, Outputs

1. **Start:** This pushbutton will be enabled once the model matrix has been fed into the GUI. This can be pressed once all the inputs needed by the GUI are completed and this would start the validation algorithm on the data and models provided. This would consider all the inputs and generate a chi-square test plots in which the calculated test scores are plotted against the prediction horizon. A threshold value is plotted as a red line. Any violation of the calculated values of the threshold means that the models fail the test for that particular prediction horizon. Calculated values below the threshold means that the model passes the validation test.

2. **Enter prediction horizon for Isolation:** The isolation is done by fixing the prediction horizon and then isolating the faulty individual model which fails the validation

test. The user can define the prediction horizon over which the isolation is desired. A guide as to which horizon to choose is got by looking at the validation output. Any prediction horizon which has the overall model failing the validation test could be used. Default is twenty five step ahead prediction (k-step and multistep).

3. **Isolation:** Once a particular model system fails the test it is important to determine which particular input's model fails the test (since we are considering MISO models as a general case). This can be done by this pushbutton. This should be used if the model fails the validation tests. This gives a bar chart showing which models are probably at fault at a scale of zero to one. The isolation algorithm assumes a fixed prediction horizon of twenty five.

4. **Reset:** Clears all GUI variables and resets all boxes to restart GUI.
   It is important to use 'reset' before running a new case study or new model set or data set.

5. **Close:** Closes GUI